Dienstplanung mit Excel

Entwurf
von Thomas Bär

Thomas Bär, Kaufbeuren

Die Deutsche Bibliothek - CIP-Einheitsaufnahme
Thomas Bär:
Dienstplanung mit Excel / Thomas Bär.
IBICURA, Unterostendorf 2004

ISBN 3-9809304-1-6

Lektorat: Ingeborg Bauer
Gestaltung: Das Projekt, Kaufbeuren
Layout und Satz: Bauer & Bauer
Druck und Verarbeitung: Druckerei Schnitzer, Marktoberdorf

01 Vorwort

Der „Dienstplaner in Excel III“ ist ein aus dem stationären Pflegedienst am Bezirkskrankenhaus Kaufbeuren hervorgegangenes Projekt zur Vereinfachung der Dienstplanung für jeweils einen Monat. Das Projekt basiert auf den Anforderungen von Stationen aus dem Bereich forensische Psychiatrie, Allgemeinpsychiatrie, Gerontopsychiatrie und dem Enthospitalisierungsbereich und wurde über mehrere Monate in Zusammenarbeit mit dem Pflegeinstitut – Bereich Informatik verfeinert und in der Leistung stark erweitert.

Es gibt viele Dienstplanlösungen mit ausgefeilten Fähigkeiten am Markt, jedoch finden sich immer wieder Argumente, die gegen die Einführung großer, professioneller Dienstplanverfahren sprechen. Die Begründungen reichen von den entstehenden Kosten für die Software und Wartungsverträge, über die fehlende Infrastruktur im Bereich der EDV, bis hin zur fehlenden Individualität, die für jede Einrichtung im Gesundheitswesen notwendig ist.

Mit der vorliegenden Version, basierend auf der Standardapplikation Excel von Microsoft, kann eine kostengünstige Dienstplanung erstellt werden, ohne große Datenbanken betreiben und warten zu müssen. Die Parametrierung, sprich die Anpassung an die eigenen Bedürfnisse kann – sofern das Standardmodell verwendet wird – innerhalb von zirka einer Stunde geschehen. Weitreichende Anpassungen an die eigenen Bedürfnisse sind durch die didaktische Aufbereitung von Funktionalitäten des Tabellenkalkulationsprogramms durch den Autor möglich. Sie erhalten mit dem Gespann von Buch und Programm das Werkzeug für eine individuelle Dienstplanung in die Hand, die ohne größere Schwierigkeiten an das eigene Unternehmen auch außerhalb des Gesundheitswesens angepasst werden kann.

Das Buch gliedert sich in insgesamt drei Teile. Der Einführung in die Tabellenkalkulation mit Excel in den ersten neun Kapiteln. Anschließend werden Anhand des Beispiels „Dienstplanung“ spezielle Funktionalitäten in Excel und Visual Basic for Applications (VBA) umgesetzt. Der dritte Teil richtet sich an den Anwender des fertigen Dienstplanmodells mit Checkliste zur Einführung, Parametrierungshilfe und einen Nutzungsvorschlag, eine Art Bedienungsanleitung.

Da es sich beim vorliegenden Verfahren um ein Werkzeug zur individuellen Anpassung handelt, kann eine rechtsverbindliche Garantie nicht gegeben werden. Eine kostenfreie Unterstützung für Probleme, die im Zusammenhang mit der Nutzung und Installation der Datei entstehen können, wird weder vom Verleger, Autoren noch vom Betreiber der Webseiten geboten. Innerhalb des Quellcodes der Datei findet sich eine ausreichende Dokumentation, die ebenfalls zur Fehlerbeseitigung dienlich sein kann. Aktuelle Informationen und neue Dateiversionen des Dienstplaners finden sich auf der Webseite des Pflegeinstituts (www.pflegeinstitut.de/dpl).

Viel Spaß bei der Dienstplanung

Der Autor

Thomas Bär wurde 1973 in Hamm in Westfalen geboren. Im Anschluss an die Realschule folgte der Besuch der Berufsfachschule für das Sozial- und Gesundheitswesen in Unna. Nach dreijähriger Lehrzeit endet die Ausbildung zum staatliche examinierten Krankenpfleger an der Krankenpflegeschule des Bezirkskrankenhauses Kaufbeuren. Anschließend folgen einige Monate Einsatz auf einer geschlossenen, psychiatrischen Station. Im Jahr 1996 und dem Januar 1997 Ableistung des Zivildienstes auf einer interdisziplinären Intensivstation im somatischen Kreiskrankenhaus der Stadt Landsberg am Lech. Anschließend Mitgründer des pps – Privaten Pflegeservices in Mauerstetten, einem ambulanten Pflegedienst. Erster Einsatz von Excel-Tabellen zur Kalkulation in der Buchhaltung, der Kosten- und Leistungsrechnung, sowie der Personalabrechnung. Seit 1999 Inhaber der Stabstelle EDV in der Pflege am Bezirkskrankenhaus Kaufbeuren. Seit 2001 Dozententätigkeit am Pflegeinstitut des Bezirks Schwaben im Bereich Informatik. Ebenfalls seit 2001 fester freier Mitarbeiter des Windows 2000 Magazin, einer unabhängigen Monatspublikation des AWi Verlag in Grasbrunn für Windows NT/2000/XP und 2003 Experten. Seit Ende 2003 stellvertretender Abteilungsleiter der EDV am Bezirkskrankenhaus Kaufbeuren.

Kurzübersicht der Kern-Funktionalitäten

- Sechs Dienstformen
- Individuelle Dienstbezeichnung
- Dienstplan auf Monatsbasis
- alle Werteingaben im Format Stunden:Minuten
- maximal 22 / 30 Mitarbeiter pro Plan
- Sollstunden für Monate lassen sich frei hinterlegen
- Feiertage lassen sich frei hinterlegen
- Definition von Sollbesetzungen auf Tages- / Wochenbasis
- Farbliche Prüfung auf korrekte Dienstbesetzung
- Farbliche Prüfung auf Einhaltung der Sollarbeitszeit der Mitarbeiter
- Sollstellung auf Basis von 100%, 75%, 50% oder 25% Anstellung
- Freie Sollstellung
- Unterscheidung von Wochenturnus- und Schichtdienst
- Farbliche Darstellung von Urlaub, Krankheit und Nachtdienst

Kurzübersicht der speziellen Funktionalitäten

- Automatische Anrechnung von Teamsitzung
- Mitarbeiterorientierte Dienstplanausgabe
- Abbildung komplett freier Dienste mit Zeitangaben
- Berechnung Nachtdienste im Quartal

Inhaltsverzeichnis

02 Grundlagen der Tabellenkalkulation

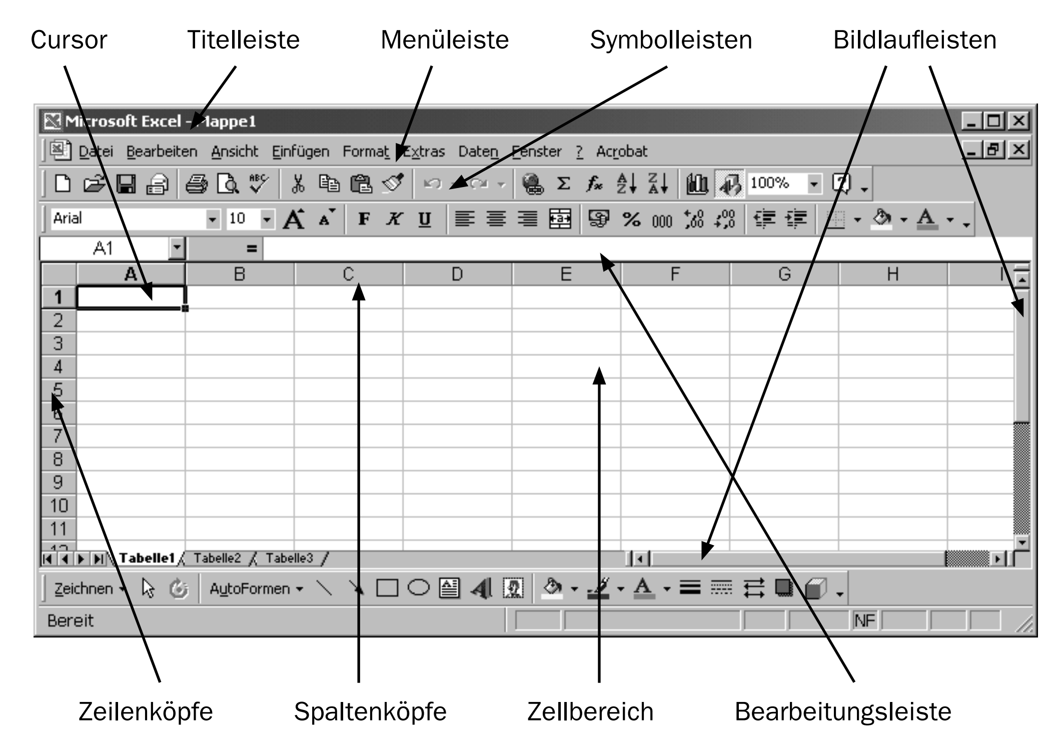
Eine Hauptaufgabe der Tabellenkalkulation mit Microsoft Excel ist der Aufbau und die Gestaltung von Tabellen und das Berechnen der erfassten Daten. Folgende Aufgaben aus dem Bereich des Pflegedienstes sind nur Beispiele die mit einer Tabellenkalkulation erarbeitet werden können:

- Rechnungsmanagement
- einfache bis komplexe Berechnungen
- statistische Erhebungen
- finanzmathematische Berechnungen
- Erstellung von Tages-, Wochen-, Monats oder Jahresplänen
- Erstellung von Dienstplänen
- Erstellung von Formularen
- Erstellung von Grafiken auf Basis von Zahlenwerten

02.1 Basics : Programmfenster von Excel

Das Microsoft Excel Dokument ist die Arbeitsmappe. Mit dem programmorientierten Start von Microsoft Excel wird eine neue Arbeitsmappe *Mappe1.xls* als Dokumentfenster geöffnet.

Einige Benennungen von Elementen unter Microsoft Excel sind für die folgenden Kapitel wichtig:



Das **Aussehen** von Microsoft Excel kann, je nach Benutzereinstellung variieren. Sind einige Symbolbefehle nicht vorhanden, so können diese im Menübefehl ANSICHT -> SYMBOLLEISTEN eingeschaltet werden. In der Standardeinstellung werden „Standard“, „Format“ und „Zeichnen“ von Microsoft Excel vorgegeben.

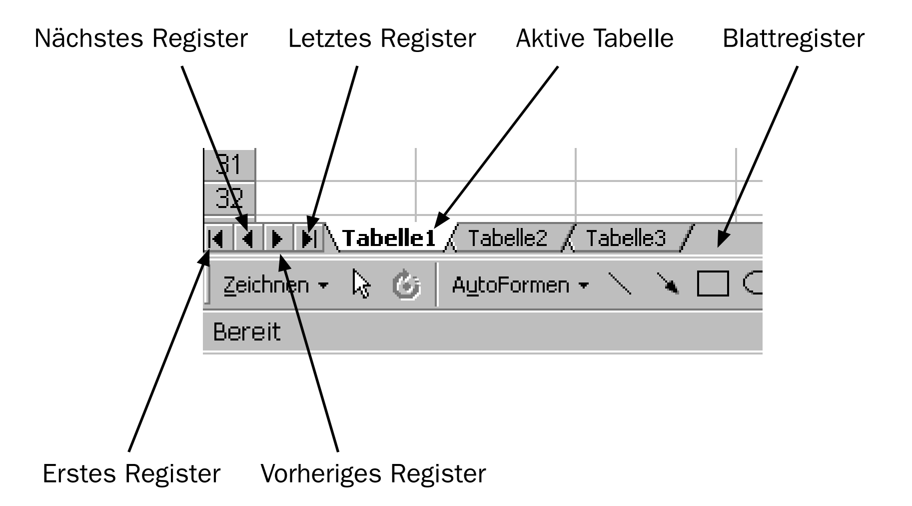
Alle Eingaben unter Microsoft Excel werden in der **Bearbeitungsleiste** vorgenommen, funktionell erinnert sie an die Browserleiste aus den Internetprogrammen.

Mit Hilfe der **Bildlaufleisten** wird der Bildausschnitt bewegt, dies ist bei größeren Tabellen erforderlich. Im **Zellbereich** werden Daten und Ergebnisse dargestellt, die hinter dem Ergebnis stehende Funktion wird erst durch die Auswahl der Zelle (Bewegen des Cursors auf die betreffende Zelle oder durch Anklicken) in der Bearbeitungsleiste sichtbar.

Durch die **Spaltenköpfe** und **Zeilenköpfe** werden Positionen innerhalb des **Zellbereiches** beschrieben. Weitere Erklärungen finden sich im Abschnitt 3 „Fast wie Schiffe versenken...“.

02.2 Basics : Register

In der Standardeinstellung des Programms enthält eine Arbeitsmappe drei leere Tabellenblätter mit dem Namen Tabelle1, Tabelle2 und Tabelle3. Der Benutzer kann diesen Tabellen (oder auch andere Arbeitsblätter z.B. Diagramme oder Visual Basic Module) weitere hinzufügen, wobei die Maximalzahl der Arbeitsblätter innerhalb einer Arbeitsmappe auf 255 begrenzt ist.



Mit einem Rechtsklick auf das Blattregister der Tabelle erscheint das Kontextmenü in dem Arbeitsblätter umbenannt, gelöscht oder hinzufügen werden können. Um die Reihenfolge der Arbeitsblätter müssen diese bei gedrückter linker Maustaste bewegt werden.

Wie die Programme Microsoft Word oder Microsoft PowerPoint prüft auch Microsoft Excel während des Öffnens einer Datei, ob diese programmierte Inhalte in Visual Basic for Applications (VBA) enthalten und erwartet vom Benutzer die Aktivierung oder Deaktivierung dieser aktiven Inhalte. Dieser Schritt ist notwendig, da Dateien mit solch aktivem Inhalt ein potentielles Sicherheitsrisiko für das System darstellen. Sollten Excel-, Word- oder PowerPoint Dateien aus unsicheren Quellen aus dem Internet bezogen werden ist die Deaktivierung dieser sog. Makros zu empfehlen.

Das Programmieren in Visual Basic for Applications (VBA) und das Erstellen von Makros zählen zu den fortgeschrittenen Möglichkeiten des Einsatzes des Microsoft Office Pakets. Im zweiten Teil dieses Buches wird auf beide Techniken näher eingegangen, da einige Funktionalitäten der Dienstplanung ohne individuellen Programmcode nicht zu realisieren sind.

02.3 Basics : Bewegung

Ein Arbeitsblatt besteht aus 256 Spalten und 65536 Zeilen. Der entstehende Schnittpunkt aus Spalte und Zeile ist eine Zelle, die durch Spaltennamen und Zeilennummer genau adressiert ist (siehe auch Abschnitt 3 „Fast wie Schiffe versenken...). Um sich in dem riesigen Bereich, der Microsoft Excel zur Verfügung stellt, schnell und effizient zu bewegen bietet das Programm mehrere Arten der **Bewegung**. Poweruser, Menschen die häufig mit Microsoft Excel arbeiten, verwenden in der Regel die **Tastatur**, da die Hände bereits auf dieser Ruhen und das „in die Hand nehmen“ der Maus eine zu große Zeit erfordern würde.

Mit der **Maus** sind Cursorbewegungen in der Tabelle sehr einfach, der Mauszeiger wird an die gewünschte Position geführt und durch einen einmaligen Klick mit der linken Maustaste wird die Zelle ausgewählt.

Für horizontale und vertikale Bewegungen werden die **Bildlaufleisten** am unteren und rechten Bildschirmrand verwendet. Für eine zeilen- beziehungsweise spaltenweise Bewegung durch die Tabelle kann der Pfeil an den Bildlaufleisten verwendet werden.

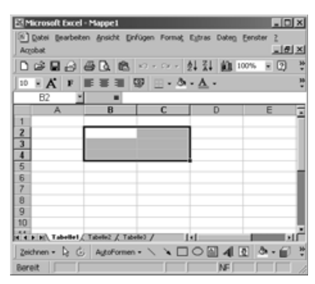
Mit der Tastatur lässt sich Microsoft Excel ebenfalls bequem steuern. Die Cursorbewegung wird mit Hilfe der Pfeiltasten herbeigeführt. Um den Bildausschnitt und andere Bewegungen durchzuführen, werden Tastenkombinationen verwendet:

STRG+ BILD AUFWÄRTS	=	nächste Tabelle im Register
STRG + BILD ABWÄRTS	=	vorherige Tabelle im Register
ALT + BILD AUFWÄRTS	=	Bildausschnittswechsel horizontal rechts
ALT + BILD ABWÄRTS	=	Bildausschnittswechsel horizontal links
BILD AUFWÄRTS	=	Bildausschnittswechsel vertikal hoch
BILD ABWÄRTS	=	Bildausschnittswechsel vertikal runter
POS1	=	Zeilenanfang
STRG + POS1	=	Tabellenanfang (Zelle A1)

02.4 Basics : Markieren

Das Markieren ist wie bei allen Programmen in grafischen Betriebssystemen, wie Microsoft Windows, eine der wichtigsten Basisfähigkeiten, die einem Benutzer abverlangt werden.

Die Markierung legt fest, auf welchen Teil im Dokument sich viele Änderungen, wie zum Beispiel Schrifttyp, Zellformat oder Farbhintergrund auswirken sollen. Mit der Maus lässt sich ein Bereich durch einen Linksklick und die Bewegung bis zur Zielzelle markieren, mit der Tastatur wird durch das gleichzeitige Drücken der Großschreibtaste (Shift) und die Bewegung des Cursors mit Hilfe der Pfeiltasten der gleiche Effekt erzielt.



02.5 Einstieg - Vorbereitungen

Im Folgenden werden notwendige Funktionen unter Microsoft Excel am Beispiel erläutert. Zwei Grundeinstellungen sind im Programm durchzuführen, um eine fehlerfreie Arbeit aller Beispiele, und des fertigen Dienstplaners zu gewährleisten.

Klicken Sie auf Extras -> Makros -> Sicherheit



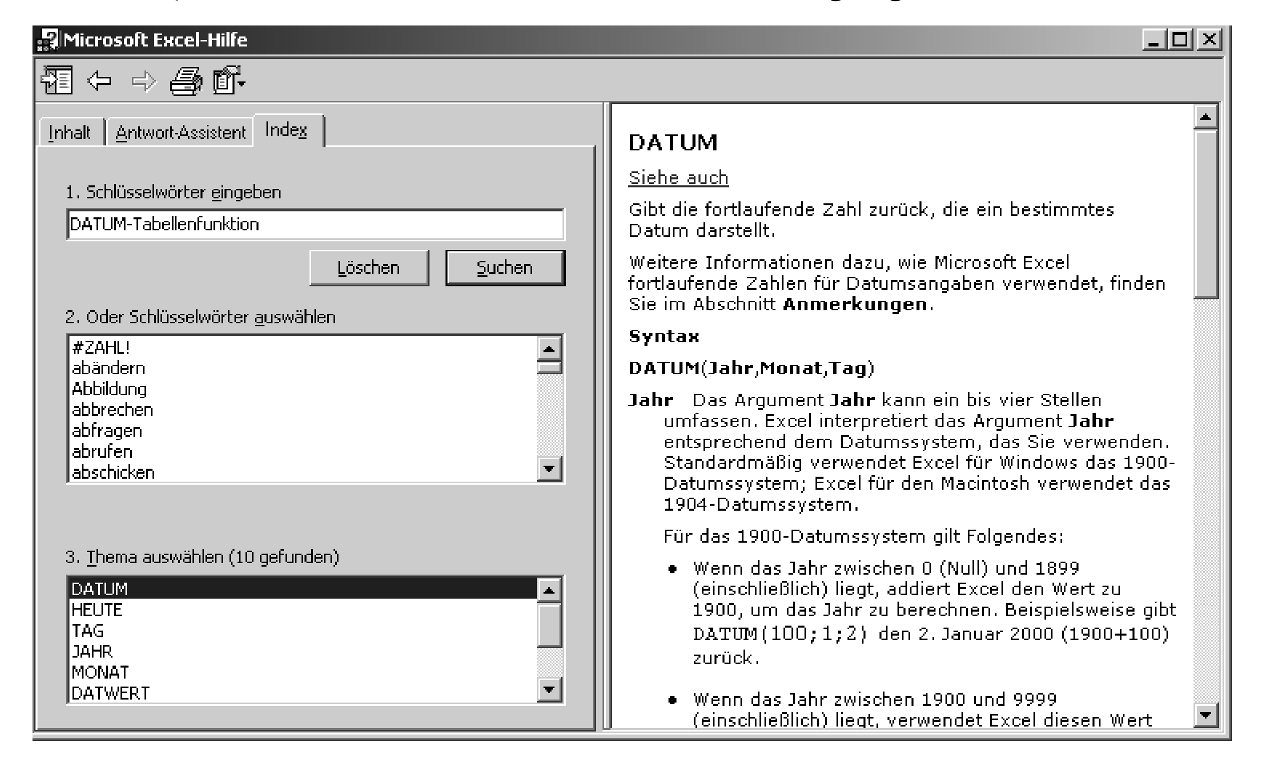
Setzen Sie den Wert auf **„Mittlere Sicherheit“**. In den hohen Sicherheitseinstellungen werden Makros und Visual Basic for Applications Programme – beides wird später noch im Detail betrachtet – deaktiviert, sodass weiterführende Techniken nicht zur Verfügung stehen. Sollten Sie nach der Festlegung der Einstellung auf „Mittel“ ein Service Pack, Hotfix oder Patch für Microsoft Office beziehungsweise Excel auf einem Rechner Einspielen, so ist der Wert automatisch wieder auf „Hoch“ eingestellt. Hintergrund dieses Verhaltens ist der Versuch vom Hersteller Microsoft, die Applikationssicherheit gegenüber Trojanern und Programm-Würmern aus dem Internet zu erhöhen.

Klicken Sie auf Extras -> Optionen -> Register „Berechnung“

Aktivieren Sie die Checkbox für **„1904-Datumswerte"**. Diese Einstellung ist erforderlich um mit Microsoft Excel auch negative Stundenwerte berechnen zu können. Die Datumsberechnung in Excel beginnt standardmäßig mit dem 01.01.1900 um 00:00 – dieser Zeitwert wird als 0 bezeichnet. Ist es 18:00 Uhr Abends am 01.01.1900 so wird dieser Zeitwert in Excel als 0,75 bezeichnet. Es werden also alle Datums- und Zeitangaben mit dem 01.01.1900 in Beziehung gestellt. Tippen Sie den Uhrzeitwert 06: 00 ein, so wird intern die Zahl 0,25 zu Berechnung verwendet. Die Datums- und Zeitangaben sind interpretierte Werte, die nur für den Menschen in ein für den Sprachraum übliches Anzeigeformat umformatiert wird. Für eine Dienstplanung oder Stundenabrechnung ist dieser Umstand von Interesse. Wird die Checkbox „1904-Datumswerte" aktiviert, so wird der 0-Wert auf den 01.01.1904 gesetzt, sodass der 31.12.1903 00:00 Uhr als Zahlenwert -1 berechnet werden kann. Datumswerte vor dem 01.01.1900 können in Excel generell nicht berechnet werden.

02.6 Hinweise zur Onlinehilfe

Aufgrund des speziellen Themas können die Funktionalitäten in diesem Buch nur gestreift werden. Da aber jedes Microsoft Excel sein eigenes Handbuch quasi mitbringt, können weiterführende Informationen direkt aus der **Onlinehilfe** entnommen werden. Nehmen wir das Thema „Datumswerte". Drücken Sie, während Sie mit Excel arbeiten, die F1 (Funktionstaste 1), so wird – gemäß den Programmierempfehlungen für Microsoft Windows – die Onlinehilfe des Programms geöffnet. Befinden Sie sich aktuell auf dem Desktop von Windows, so würde die Onlinehilfe für Windows angezeigt werden.



Im ersten Register der Onlinehilfe „Inhalt" werden durch kleine lilafarbene Bücher einzelne Themenbereiche symbolisiert. Register „Antwort Assistent" erwartet eine komplett formulierte Frage und versucht entsprechend von Schlüsselwörtern in Ihrer Fragestellung für die Antwort relevante Themenkomplexe aus der Onlinehilfe einzublenden. Das letzte Register „Index" ist eine alphabetische Aufzählung aller Themenbereiche. Wird dort das Wort „Datum..." eingegeben so ist schnell erkennbar, welche Informationen die Onlinehilfe zum Thema „Datum" bereithält. Doppelklicken Sie das Schlüsselwort und verwandte Themen und die Hauptbeschreibung zum Schlüsselwort werden angezeigt.

Ebenfalls sehr hilfreich bei der Suche nach den richtigen Parametern eines Excel Befehls ist der Befehl „**Funktionen einfügen**". Sie erreichen diesen Assistenten über Einfügen -> Funktion. In der linken Hälfte des kleinen Fensters werden die Kategorien der Befehle angezeigt, auf der rechten Seite die in der Rubrik vorhandenen Befehle. Unterhalb des Auswahlbereichs findet sich ein kurzer Erklärungssatz, der bei der Suche nach einer speziellen Funktion behilflich sein kann. Nach einem Doppelklick auf die Funktion haben Sie die Möglichkeit innerhalb des grafischen Fensters die Parameter des Befehls einzutragen oder mit dem Mauszeiger Bereiche und Werte auszuwählen. Das Praktische an diesem Assistenten ist, dass das Ergebnis der Funktion bereits im Programmfenster zu sehen ist.

03 Fast wie Schiffe versenken...

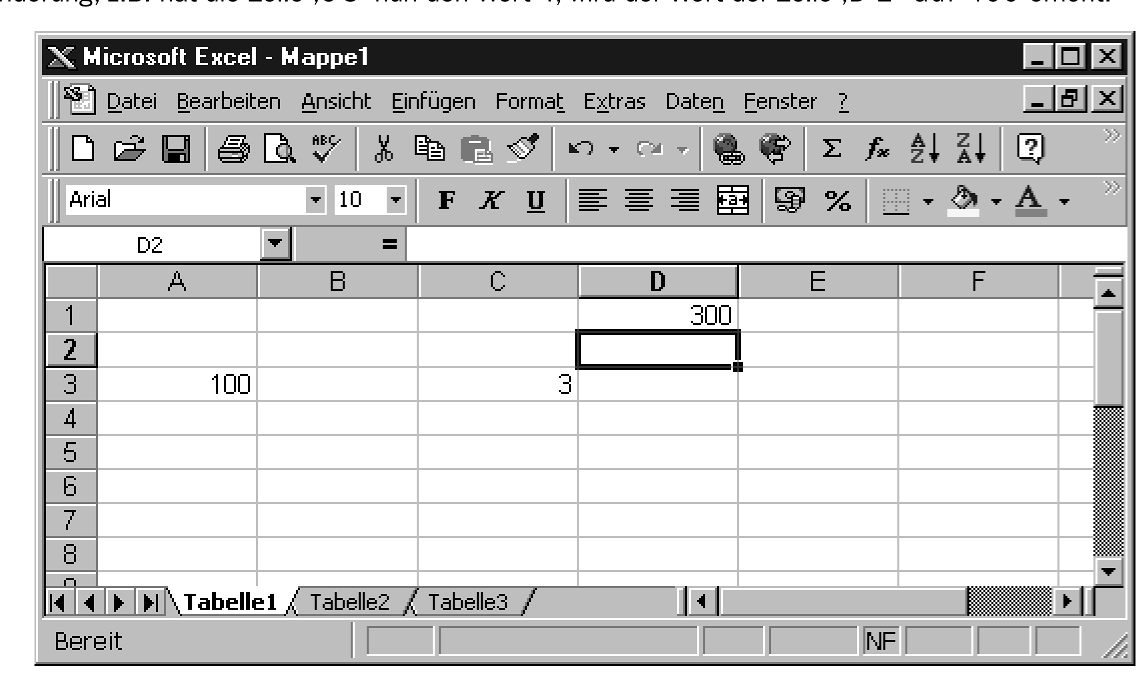
Jeder von Ihnen dürfte mit den Regeln des Spiels ‚Schiffe versenken' vertraut sein – durch die Angaben von vertikalen und horizontalen Positionswerten wird versucht des Gegners Schiffe zu treffen. Zwischen diesem Spielsystem und der Arbeitsweise einer Tabellenkalkulation gibt es gewisse **Parallelen**. Es werden bei der Tabellenkalkulation zwar keine Schlachtschiffe versenkt, aber in gleicher Art und Weise Zellen angesprochen.

Gehen wir davon aus, dass eine Zelle mit dem Namen ‚A-3' den Wert 100 besitzt und eine zweite Zelle mit dem Namen ‚C-3' den Wert 3. Wird nun eine Berechnung angefordert nach dem Muster:

Multipliziere den Wert der Zelle ‚A-3' mit dem Wert der Zelle ‚C-3'

dürfte als Ergebnis 300 zu erwarten sein. Wird diese Rechenoperation in einer Zelle mit dem Namen, D-1' durchgeführt hat diese nach der Bestätigung der Eingabe mit der Eingabetaste den Wert 300.

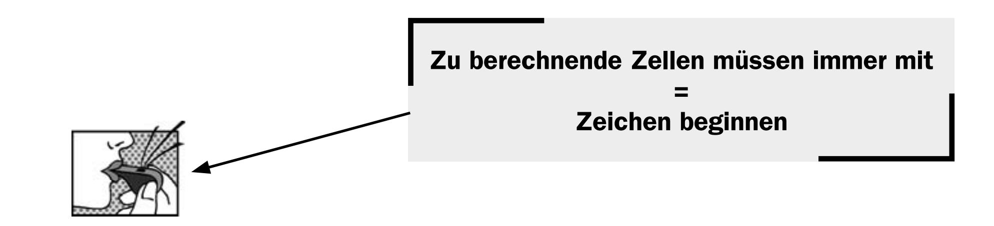
Diesen Wert behält Sie bis sich die Basiswerte in ‚A-3' oder ‚C-3' verändern – sofort nach einer Veränderung, z.B. hat die Zelle ‚C-3' nun den Wert 4, wird der Wert der Zelle ‚D-1' auf 400 erhöht.



Dieses **erste Beispiel** wird nun in Microsoft Excel eingegeben:

Tippen Sie in die Zelle A3 den Wert 100 ein, bestätigen Sie die Eingabe mit der Eingabetaste. In die Zelle C3 wird der Wert 3 eingetragen und ebenfalls mit der Eingabetaste bestätigt (Sie müssen jede Eingabe im Programm mit der Eingabetaste bestätigen, um die Eingabe zu bestätigen). Der eingegebene Wert wird unterhalb der Menüleiste im Feld „=" angezeigt, bis die Eingabetaste gedrückt wurde.

Die Berechnung „Multipliziere Wert aus A3 mit C3" wird als so genannte Funktionen in das Feld D1 eingeben. Funktionen müssen immer mit einem „Ist gleich" Zeichen begonnen werden, daran erkennt das Tabellenkalkulationsprogramm, das es sich um zu berechnende Werte handelt, und nicht um Eingaben.



Tippen Sie in die Zelle D1 folgende Formel ein:

= A3 *C3

Nach dem Drücken der Eingabetaste erscheint in der Zelle erwartungsgemäß das Ergebnis: 300.

Ändern Sie beispielsweise einen Wert in A3 oder C3, so wird der kalkulierte Wert in D1 automatisch neu berechnet. In einem nächsten Schritt werden nun die weiteren drei Grundrechenarten in Microsoft Excel eingesetzt.

Tippen Sie folgende Funktionen ein:

in D2 = D1+20

in D3 = D2 / 5

und in D4 = D2 - D3

Wie an den Funktionen erkennbar ist, lassen sich auch einfache Zahleneingaben mit den berechneten Werten in der Schriftweise Spalte/Zeile (beispielsweise D2) mischen. Der Querstrich (über der Taste 7, oder als direkte Taste im numerischen Block der erweiterten PC- Tastatur) übergibt den Divisionsbefehl an Microsoft Excel.

03.1 Mehrere Zellen summieren

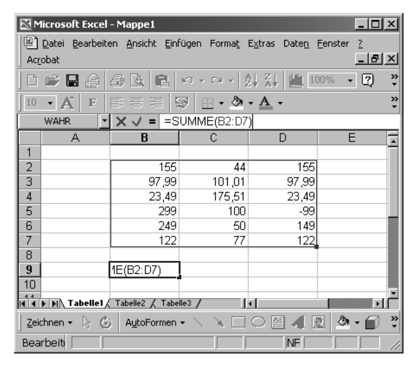
Soll ein größerer Zellbereich bearbeitet werden, ist die Eingabe der einzelnen Zellpositionen ein **mühseliges** Unterfangen. Beim Schreiben einer Funktion in der Bearbeitungszeile ist es möglich anstelle der direkten Zellposition die Maus zur Hilfe genommen werden und auf die Zielzelle geklickt werden.

Sollen alle Werte in der obigen Abbildung addiert werden müsste in B8 folgende Funktion eingegeben werden:

= B2 + B3 + B4 + B5 + B6 + B7

Da dies bei sehr großen Tabellen kaum möglich ist, bietet sich der Befehl SUMME an, der den gleichen Effekt erzielt:

= SUMME (B2:B7) „Bilde Summe aus Bereich B2 bis B7“



Das funktioniert auch bei Zellen im größeren Rahmen:

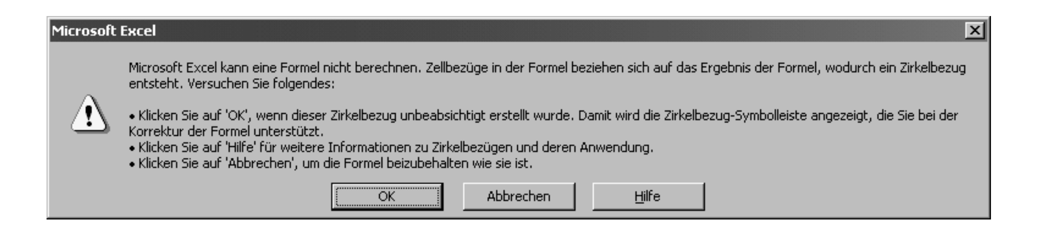
= SUMME (B2:D7) „Bilde Summe aus Bereich B2 bis B7“

Um eine ganze Spalte zu summieren kann auch die horizontale Positionsangabe ganz weggelassen werden:

= SUMME (B:B) „Bilde Summe aller Zellen in der Spalte B“

03.2 Wenn es mal schräg wird – Fehler

Wird das letzte Beispiel aus Abschnitt 3.1 in der Zelle B8 eingegeben, so erscheint folgende Fehlerdialog von Microsoft Excel:



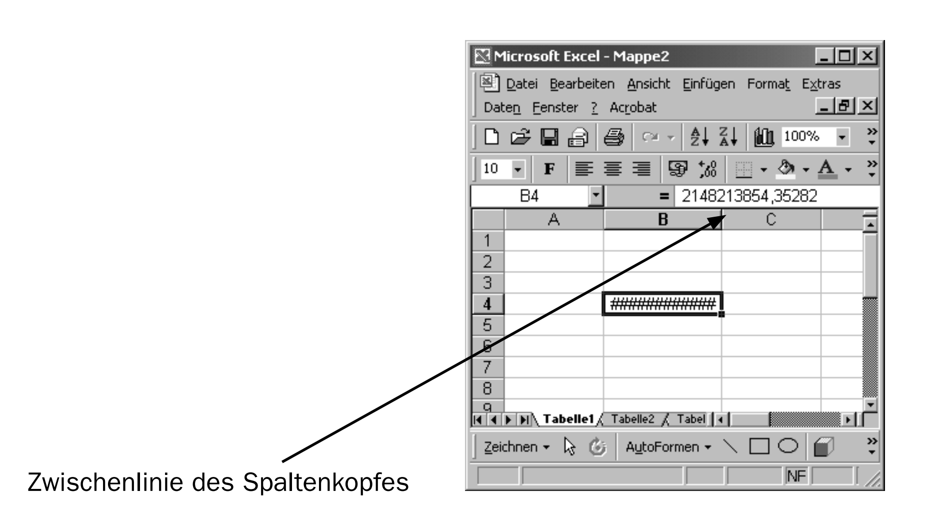
Da die Ergebniszelle (B8) selbst Teil der Spalte B ist, kann das Ergebnis von Microsoft Excel nicht gebildet werden, da sich der Wert B8 durch die Berechnung ändert und die Neukalkulation des gesamten Arbeitsblattes anfordert, was wieder eine Neuberechnung zur Folge hat, mit der Folge einer **Neuberechnung** usw. Sprich, Microsoft Excel kann diesen Befehl niemals fertig stellen – der PC würde „hängen bleiben“, da eine unmöglich zu lösende Aufgabe ansteht.

Wird jedoch in der Zelle A8 der Befehl

= SUMME (B:B)

eingegeben, so funktioniert die Funktion problemlos.

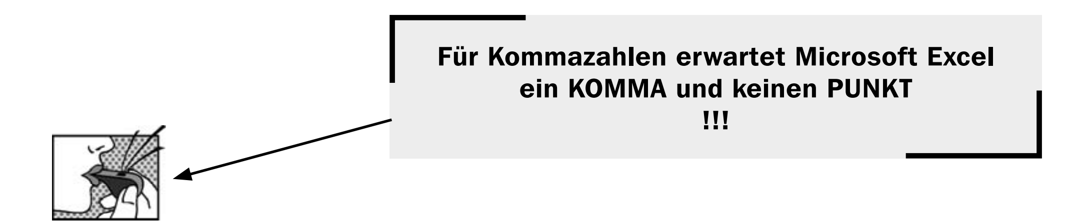
Ein deutlich häufiger auftretender Fehler ist der im Volksmund scherzhaft genannter „Gartenzaun“:



Die Spaltenbreite von B reicht nicht aus um die Zahl 2148213854,35282 darzustellen, aus diesem Grund produziert Microsoft Excel eine Reihe Doppelkreuze. Um die Spalte B zu verbreitern können Sie entweder die Zwischenlinie im Spaltenkopf klicken und verbreitern oder durch einen Rechtsklick auf den Spaltenkopf das Kontextmenü mit der Spaltenbreite zum Vorschein bringen.
Ein Doppelklick auf die Zwischenlinie formatiert die Spalte automatisch auf die notwendige Breite!

03.3 Kommastellen

So wie das Beispiel nun vorliegt, geben alle nachfolgenden Berechnungen immer einen ganzen Zahlenwert, ohne Komma aus. Würden Sie in A3 eine 103,29 eingeben, so würde in D4 der Wert 263,896 erscheinen.

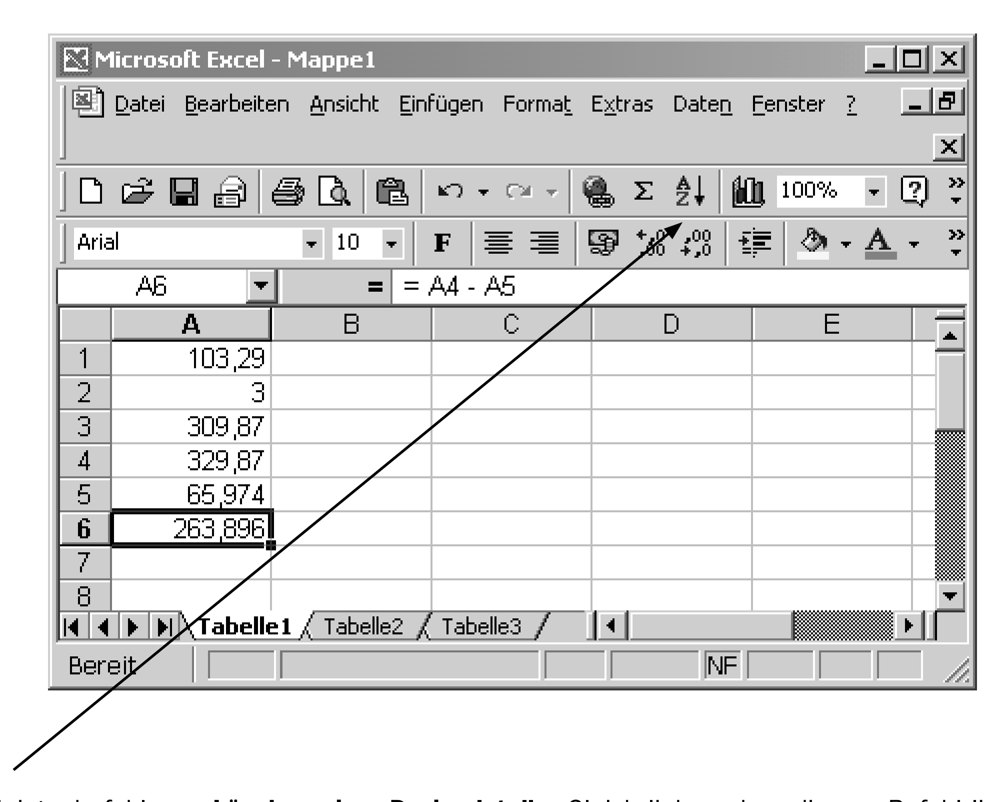


Folgende Überlegung gilt es nun bezüglich des Kommas zu machen:

a.) Wollen Sie beispielsweise einfach nur zwei Stellen hinter dem Komma sehen, jedoch Excel dazu bringen weiterhin mit allen Stellen hinter dem Komma zu rechnen

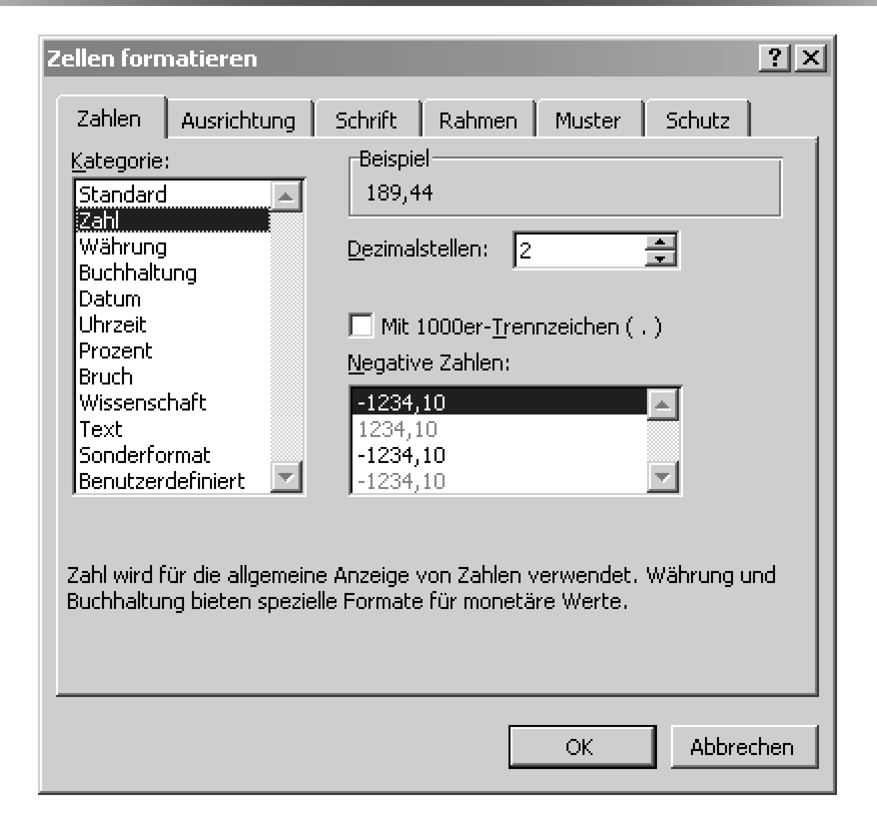
b.) Soll hinter dem Komma beispielsweise nur zwei Stellen hinter dem Komma haben, die auch für die darauf aufsetzenden Berechnungen zu verwenden sind

Die Möglichkeit A lässt sich auf zweierlei Weg herstellen:



Symbolleistenbefehl zum **Löschen einer Dezimalstelle.** Gleich links neben diesem Befehl liegt der Befehl um eine Dezimalstelle hinzufügen – auch wenn dies eventuell die Ausgabe „,0" zur Folge hätte.

Die zweite Methode ist der Einstellung im Formatierungsfenster für Zellen unter Microsoft Excel. Klicken Sie dazu auf FORMAT -> ZELLEN:



Wählen Sie in der Kategorie „Zahl" und es lassen sich in der rechten Fensterhälfte die Dezimalstellen hinter dem Komma bestimmen. Ist mehr als eine Zelle im Vorfeld ausgewählt worden, so wirken sich diese Format- Änderungen auf den gesamten, ausgewählten Zellbereich aus.

Microsoft Excel versucht bereits während der Eingabe das vom Benutzer gewünschte **Zellformat** festzulegen, so wählt Microsoft Excel bei der Eingabe von 05.05.2002 automatisch als Format ein Datum aus und wandelt die Eingabe intern auf einen berechenbaren Zellinhalt.

Wird nun anschließend in diese Zelle ein normaler Zahlwert eingegeben, so würde auch dieser wieder als Datumsformat dargestellt werden. In diesem Fall ist der Benutzer gezwungen das Format des Zellbereiches mit Hilfe dieses Fenster auf die gewünschte Darstellung umzustellen.



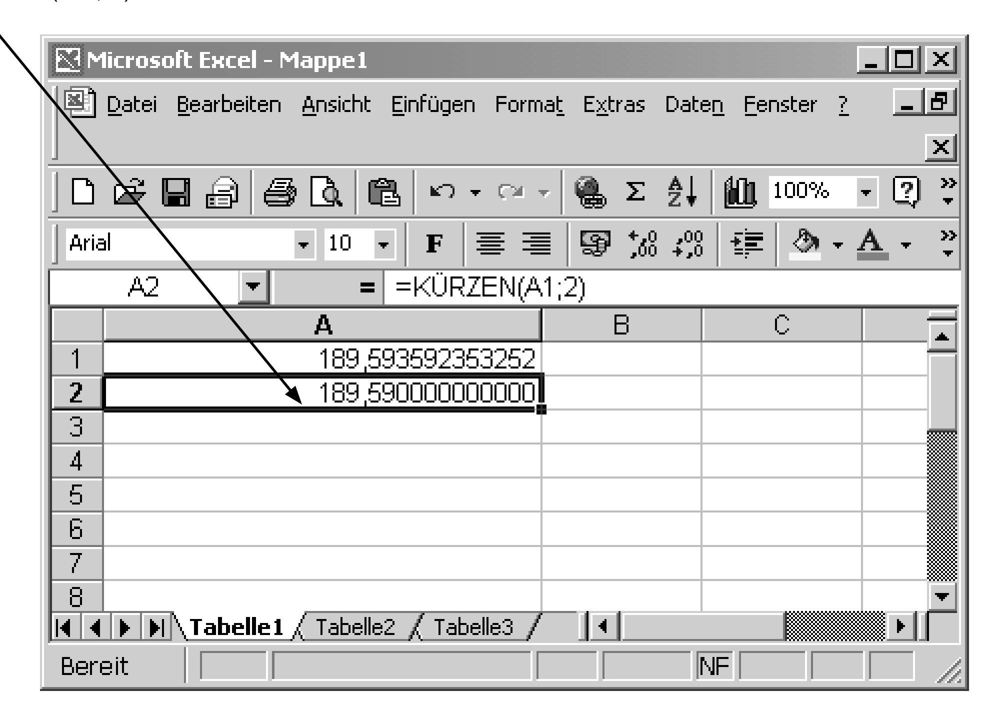
Unter Methode „B" wurde das korrekte mathematische Kürzen einer Dezimalstelle als Möglichkeit eines gleichmäßigen Erscheinungsbildes für Zahlen genannt. Während sich die zwei Lösungsansätze bisher auf das Ausblenden von Dezimalstellen bezogen, wird bei diesem Ansatz die Kommazahl wirklich entsprechend gekürzt. Das heißt auf diesen Zellwert aufbauende Berechnungen arbeiten mit dem gekürzten Wert, statt dem bedeutend korrekteren Wert, den Microsoft Excel ursprünglich berechnet hat.

Der für das Kürzen benötigte Befehl lautet:

= KÜRZEN (Wert oder Zellenangabe **;** Stellen hinter dem Komma**)**

Beispielsweise :

= KÜRZEN (A1;2)



Würden die in der Abbildung gezeigten Zahlenwerte jeweils mit dem Faktor 1000 malgenommen werden, so käme es zu einer Abweichung von 3,59235325 – einem Wert der für Berechnungen in der Buchhaltung indiskutabel ist.

Sofern es nicht, mathematisch unbedingt gewünscht ist, sollte dem Ausblenden von Dezimalstellen vor dem Kürzen einer Zahl immer der Vorrang gegeben werden, um möglichst exakte Ergebnisse zu erzielen. Dies gilt insbesondere dann, wenn mit Stundenwerten gerechnet wird, da Microsoft Excel die Stunden und Minuten als Bruchteil eines ganzen Tages verarbeitet.

03.4 Tipps zur Eingabe von Zahlen

Zeichen, die als Zahlen verwendet werden können Bei Microsoft Excel kann eine Zahl nur die folgenden Zeichen enthalten:

0 1 2 3 4 5 6 7 8 9 + - ( ) , / $ % . E e

Vorangestellte Pluszeichen (+) werden ignoriert, ein einzelnes Komma gilt als Dezimalkomma. Alle anderen Kombinationen aus Zahlen und nicht numerischen Zeichen gelten als Text.

Eingabe von Brüchen
Damit ein Bruch nicht als Datum eingegeben wird, stellen Sie bei Brüchen eine 0 (Null) voran: geben Sie beispielsweise 0 1/2 ein.

Eingabe von negativen Zahlen
Stellen Sie negativen Zahlen ein Minuszeichen (-) voran, oder setzen Sie die Zahlen in Klammern ().

Zahlenausrichtung Alle Zahlen in einer Zelle werden rechtsbündig angezeigt. Um die Ausrichtung zu ändern, markieren Sie die Zellen. Klicken Sie im Menü Format auf Zellen, und wählen Sie dann in der Registerkarte Ausrichtung die gewünschten Optionen aus.

Anzeige von Zahlen
Das für eine Zelle geltende Zahlenformat bestimmt, wie Excel eine Zahl in einem Tabellenblatt anzeigt. Bei der Eingabe einer Zahl in eine Zelle mit dem Zahlenformat Standard kann Excel ein anderes Zahlenformat anwenden. Wenn Sie z. B. 14,73 € eingeben, wendet Excel ein Währungsformat an. Um das Zahlenformat zu ändern, markieren Sie die Zellen, welche die Zahlen enthalten. Im Menü Format klicken Sie auf Zellen, und wählen Sie dann in der Registerkarte Zahlen eine Kategorie und ein Format aus.

Standard-Zahlenformat
In Zellen, denen das Zahlenformat Standard zugewiesen wurde, zeigt Excel Zahlen als ganze Zahlen (789), Dezimalbrüche (7,89) oder in wissenschaftlicher Schreibweise (7,89E+08) an, wenn die Zahl länger ist als die Zellenbreite. Im Format Standard werden bis zu 11 Ziffern angezeigt, Dezimalkomma sowie Zeichen wie „E“ und „+“ eingeschlossen. Um mit Zahlen zu arbeiten, die mehr als 11 Ziffern enthalten, können Sie ein vordefiniertes wissenschaftliches Zahlenformat (exponentielle Notation) oder ein benutzerdefiniertes Zahlenformat verwenden.

Begrenzung auf 15 Ziffern
Unabhängig von der Anzahl der angezeigten Ziffern speichert Excel Zahlen mit einer Genauigkeit von bis zu 15 Stellen hinter dem Komma. Besteht eine Zahl aus mehr als 15 signifikanten Ziffern, wandelt Excel die übrigen Stellen in Null (0) um.

Eingabe von Zahlen als Text
Excel speichert Zahlen als numerische Daten, auch wenn Sie den Zellen, die Zahlen enthalten, über den Befehl Zellen das Format Text zugewiesen haben. Damit Excel Zahlen wie z. B. Artikelnummern als Text interpretiert, weisen Sie den leeren Zellen zuerst das Format Text zu, und geben Sie dann die Zahlen ein. Haben Sie die Zahlen bereits eingegeben, weisen Sie den Zellen das Format Text zu, klicken Sie auf jede Zelle, drücken Sie F2 und dann die EINGABETASTE, um die Daten erneut einzugeben.

Ländereinstellungen
Welche Zeichen als Zahlen erkannt werden, hängt von den Optionen ab, die Sie unter „Ländereinstellungen“ in der „Systemsteuerung“ gewählt haben. Die gewählten Einstellungen bestimmen auch das Standard-Zahlenformat - beispielsweise den Punkt (.) als Dezimalpunktzeichen bei Systemen, die sich an US-amerikanischen Konventionen orientieren.

03.5 Tipps zur Eingabe von Datums- und Zeitangaben

Datums- und Zeitangaben als Zahlen
Bei Microsoft Excel gelten Datums- und Zeitangaben als Zahlen. Die Anzeige von Uhrzeit oder Datum im Tabellenblatt richtet sich nach dem Zahlenformat der Zelle. Bei der Eingabe eines Datums oder einer Uhrzeit, die Excel erkennt, wechselt das Zellformat vom Zahlenformat **Standard** zu einem vordefinierten Datums- oder Uhrzeitformat. Standardmäßig werden Datum und Uhrzeit in einer Zelle rechtsbündig ausgerichtet. Kann Excel das Datums- oder Uhrzeitformat nicht erkennen, wird das Datum bzw. die Uhrzeit als Text eingegeben, der in der Zelle linksbündig angeordnet ist.

Systemsteuerung
Die Optionen, die Sie bei den Ländereinstellungen in der Systemsteuerung auswählen, bestimmen das Standardformat für das aktuelle Datum und die aktuelle Uhrzeit sowie die Zeichen, die als Trennzeichen für Datum und Uhrzeit erkannt werden, z. B. den Schrägstrich (/) und den Gedankenstrich (-) für Datumsangaben und den Doppelpunkt (:) für Zeitangaben bei Systemen, die sich an US-amerikanischen Konventionen orientieren.

Verwenden Sie eine 4-zahlige Jahresangabe
Wenn Sie Datumsangaben, wie beispielsweise den ersten Dezember eingeben, wird von Excel zuerst die Tages- und anschließend die Jahresangabe angeglichen. Beispielsweise wird die Eingabe Dezember 01 als der erste Dezember des aktuellen Jahres und nicht als Dezember des Jahres 2001 interpretiert.

Eingabe von Datums- und Zeitangaben in dieselbe Zelle
Um Datums- und Zeitangaben in dieselbe Zelle einzugeben, trennen Sie Datum und Zeit durch ein Leerzeichen.

Uhrzeit im 12- oder 24-Stundenformat
Um die Uhrzeit im 12-Stundenformat einzugeben, setzen Sie ein Leerzeichen und anschließend **am** oder **pm** (bzw. **a** oder **p)** hinter die Zeitangabe. Andernfalls zeigt Excel die Uhrzeit im 24-Stundenformat an. Wenn Sie z. B. **3:00** an Stelle von **3:00 pm** eingeben, wird die Zeit als 3:00 Uhr morgens gespeichert.

Anzeigen der seriellen Zahl hinter einer Datums- oder Zeitangabe
Unabhängig von dem für die Anzeige von Datum oder Uhrzeit verwendeten Format speichert Excel alle Datumswerte als serielle Zahlen und alle Zeitwerte als Dezimalbrüche. Um ein Datum als serielle Zahl oder eine Uhrzeit als Dezimalbruch anzuzeigen, markieren Sie die Zellen, die Datum oder Uhrzeit enthalten. Klicken Sie im Menü **Format** auf **Zellen**, dann auf die Registerkarte **Zahlen** und schließlich im Feld **Kategorie** auf **Standard**.

Berechnungen unter Verwendung von Datums- und Zeitwerten
Zeit- und Datumswerte können addiert, subtrahiert und in anderen Berechnungen verwendet werden. Um einen Datums- oder Zeitwert in einer Formel zu verwenden, geben Sie den Wert als Text in Anführungszeichen ein. Die folgende Formel zeigt z. B. eine Differenz von 68 an:
=“12.5.2004“-“5.3.2004“

1904-Datumwertesystem
Bei Excel für Windows (und Lotus 1-2-3) werden Tage vom Anfang dieses Jahrhunderts an gezählt; die serielle Datumszahl 1 entspricht dem Datum 1. Januar 1900. Excel für den Macintosh verwendet das 1904-Datumswertesystem; die serielle Datumszahl 1 entspricht dem Datum 2. Januar 1904. Um das Datumssystem für Berechnungen zu ändern, klicken Sie im Menü **Extras** auf **Optionen** und dann auf die Registerkarte **Berechnen**. Unter **Arbeitsmappe** aktivieren Sie das Kontrollkästchen **1904-Datumswerte**. Verwenden Sie das 1904-Datumswertesystem für eine Arbeitsmappe, wenn Sie diese zusammen mit Arbeitsmappen nutzen, in denen das 1904-Datumswertesystem eingesetzt wird. Gleichzeitig ermöglicht die Aktivierung des 1904-Datumwertesystems die Berechnung negativer Zeitwerte.

Wenn Sie in Excel für Windows eine Datei öffnen, die in Excel, Version 2.0 oder höher, für den Macintosh erstellt wurde, erkennt Excel das Dateiformat und ändert die Datumswerte automatisch in das 1900-Datumswertesystem. Ebenso wird eine Datei, die mit Excel für Windows erstellt wurde, beim Macintosh in das 1904-Datumswertesystem umgewandelt.

03.6 Datums- und Uhrzeitangaben in Microsoft Excel

Microsoft Excel speichert Datumsangaben als so genannte fortlaufende Zahlen. Excel speichert Uhrzeitangaben als Dezimalbrüche, da diese als Bruchteile eines Tages angesehen werden können. Datums- und Uhrzeitangaben sind Werte und können daher addiert, subtrahiert und in anderen Berechnungen verwendet werden. Um beispielsweise den Abstand zwischen zwei Datumsangaben zu ermitteln, können Sie die Daten voneinander subtrahieren. Sie können eine Datumsangabe als fortlaufende Zahl und eine Uhrzeitangabe als Dezimalbruch anzeigen, indem Sie das Format der Zelle, die das Datum oder die Uhrzeit enthält, in das Format Standard ändern.

Excel unterstützt zwei Datumswertesysteme: das 1900- und das 1904-Datumswertesystem. In Microsoft Excel für Windows wird standardmäßig das 1900-Datumswertesystem verwendet. Um zum 1904-Datumswertesystem zu wechseln, klicken Sie im Menü Extras auf Optionen, klicken Sie auf die Registerkarte Berechnung, und aktivieren Sie dann das Kontrollkästchen 1904-Datumswerte.

Die folgende Tabelle enthält für beide Datumswertesysteme das erste und das letzte Datum sowie die entsprechenden seriellen Datumswerte.

Datumswertesystem	Erstes Datum	Letztes Datum
1900	1. Januar 1900 (serieller Wert 1)	31. Dezember 9999 (serieller Wert 2958465)
1904	2. Januar 1904 (serieller Wert 1)	31. Dezember 9999 (serieller Wert 2957003)

Angabe des Jahres durch zwei Ziffern
Wenn Sie ein Datum in eine Zelle eingeben und nur zwei Ziffern verwenden, interpretiert Excel die Jahresangabe folgendermaßen:

als Jahreszahl des Zeitraums 2000 bis 2029, wenn Sie für das Jahr die Werte 00 bis 29 eingeben. Beispielsweise interpretiert Excel die Eingabe 28.05.19 als den 28. Mai 2019.

als Jahreszahl des Zeitraums 1930 bis 1999, wenn Sie für das Jahr die Werte 30 bis 99 eingeben. Beispielsweise interpretiert Microsoft Excel die Eingabe 28.05.98 als den 28. Mai 1998.

Angabe des Jahres durch vier Ziffern
Damit sichergestellt ist, dass Jahresangaben wie beabsichtigt interpretiert werden, geben Sie die Werte mit vier Ziffern ein (2001 anstatt 01). Wenn Sie vier Ziffern eingeben, muss Excel nicht das Jahrtausend für Sie interpretieren.

Die Tabellenfunktion DATUM
Wenn Sie einen Teil einer Datumsangabe, beispielsweise das Jahr oder den Monat, in einer Formel bearbeiten müssen, können Sie die Tabellenfunktion DATUM verwenden.

Die Tabellenfunktion ZEIT
Wenn Sie einen Teil einer Uhrzeitangabe, beispielsweise die Stunde oder Minute, in einer Formel bearbeiten müssen, können Sie die Tabellenfunktion ZEIT verwenden.

Standardmäßiges Anzeigen vierstelliger Jahreszahlen
Standardmäßig werden Datumsangaben bei der Eingabe in einer Arbeitsmappe mit zweistelliger Jahresanzeige formatiert. Sie können mit der Windows-Systemsteuerung das Standarddatumsformat ändern, so dass Jahresangaben vierstellig anstatt zweistellig angezeigt werden.

03.8 Tipps zur Eingabe von Texten

Bei Microsoft Excel gilt jede beliebige Kombination von Zahlen, Leerzeichen und nichtnumerischen Zeichen als Text, wie beispielsweise:
10AA109, 127AXY, 12-976, 208 4675.

Der gesamte Text wird in einer Zelle linksbündig ausgerichtet. Zum Ändern der Ausrichtung klicken Sie im Menü Format auf Zellen, und wählen Sie dann in der Registerkarte Ausrichtung die gewünschten Optionen aus.

Um den gesamten in der Zelle enthaltenen Text in mehreren Zeilen anzeigen zu lassen, aktivieren Sie in der Registerkarte Ausrichtung das Kontrollkästchen Zeilenumbruch.

Um einen manuellen Zeilenwechsel in eine Zeile einzufügen, drücken Sie ALT+EINGABE.

Soll beispielsweise der Bestellcode 000482348 eingegeben werden, so würde Microsoft Excel die Eingabe automatisch als Zahl erkennen und die drei führenden Nullangaben ignorieren und stattdessen 482348 in die Tabelle schreiben. Durch die Eingabe eines Hochkommas (oberhalb des Doppelkreuzes auf der Tastatur neben der zentralen Eingabetaste) wird Microsoft Excel informiert, dass die folgende Eingabe exakt so darzustellen, wie sie vom Benutzer eingegeben wurde.

03.9 Weitere Eingabehinweise

Eingabetastenverhalten :
Um festzulegen, welche Zelle nach Drücken der EINGABETASTE zur aktiven Zelle wird, klicken Sie im Menü Extras auf Optionen und dann auf die Registerkarte Bearbeiten.

Um zu einer angrenzenden Zelle zu springen, aktivieren Sie das Kontrollkästchen Markierung nach dem Drücken der Eingabetaste verschieben, und wählen Sie dann im Feld Extrusionsrichtung eine Richtung aus. Soll die aktuelle Zelle aktiv bleiben, deaktivieren Sie das Kontrollkästchen.

Automatische Vervollständigung :
Klicken Sie im Menü Extras auf Optionen und dann auf die Registerkarte Bearbeiten.

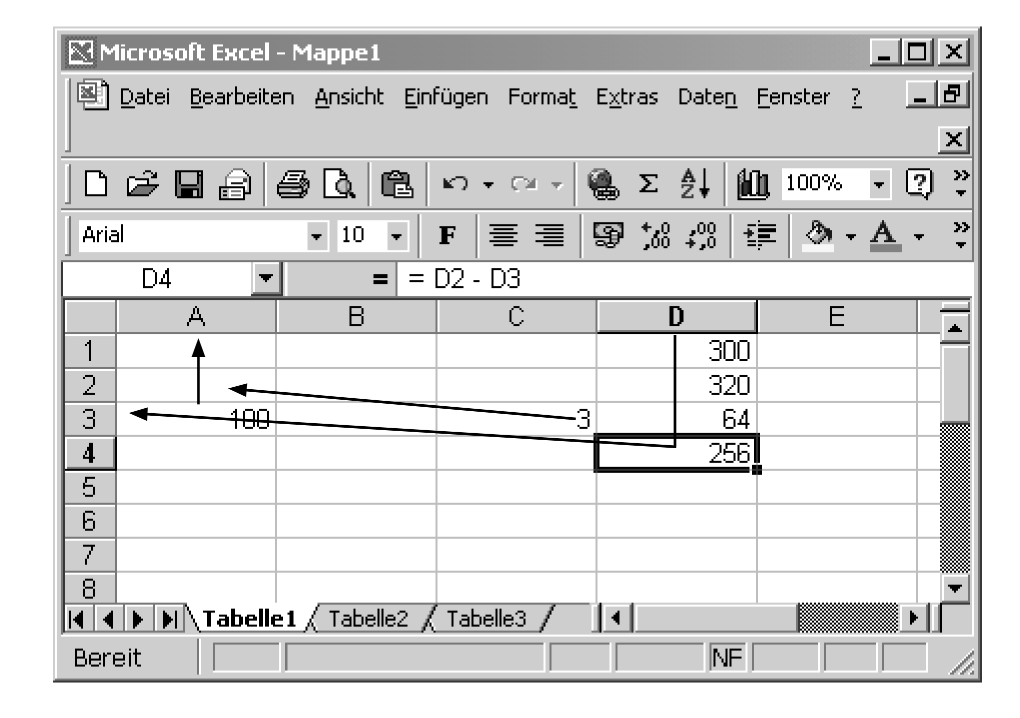
Um die automatische Vervollständigung von Einträgen zu ermöglichen, aktivieren Sie das Kontrollkästchen AutoEingabe für Zellwerte aktivieren. Um die automatische Vervollständigung zu unterdrücken, deaktivieren Sie dieses Kontrollkästchen. Siehe auch Abschnitt 5.3 für weitere Informationen.

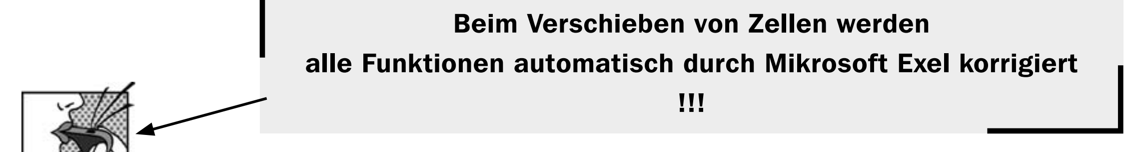
04 Zellen komplett bewegen

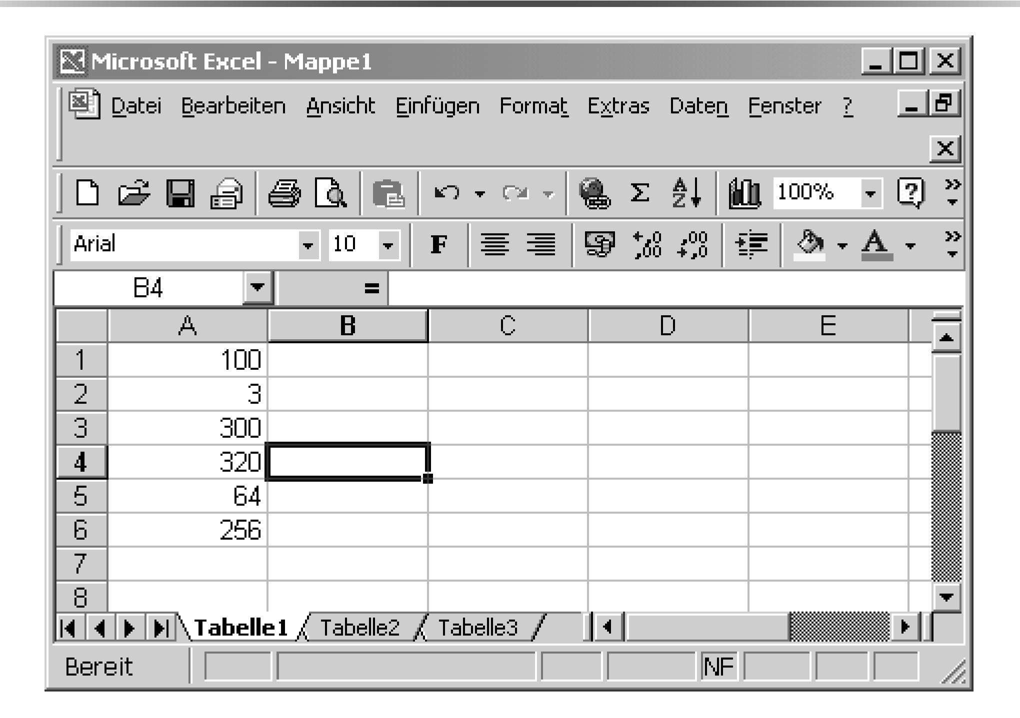
Manchmal ist es notwendig eine Tabelle neu zu gestalten. Damit eine solche Neugestaltung nicht die komplette Neuanlage einer Tabelle zur Folge hat, können Zellen bewegt werden.

Dazu wird auf den schwarzen Rahmen einer Zelle exakt geklickt, die linke Maustaste bleibt weiter gedrückt (Befehlssequenz: Ziehen) und die ganze Zelle an einen neuen Zielort bewegt. Microsoft Excel ändert nach der Bewegung alle betroffenen Funktionen dahingehend um, dass diese weiter korrekt rechnen.

Soll ein ganzer Block von Zellen bewegt werden, so wird dieser Bereich zunächst markiert (die erste Zelle wird bei diesem Vorgang üblicherweise nicht bläulich hinterlegt und erhält nur einen Rahmen). Anschließend ist an einer Stelle des Rahmens genauso vorzugehen, als ob lediglich eine einzige Zelle bewegt werden muss.
Bewegen Sie zum Training nach folgendem Beispiel:







Wie Sie erkennen werden tatsächlich alle Funktionen durch Microsoft Excel **umgeschrieben.** Sollte bei diesem Training ein Versuch einmal negativ enden, so können Sie über die Befehlsfolge BEARBEITEN -> RÜCKGÄNGIG den eben durchgeführten Schritt zurücknehmen.

Einfacher lässt sich die Rücknahme jedoch über die Tastaturabkürzung STRG und Z (beide Tasten gleichzeitig drücken) herbeiführen. Wurde ein Schritt zuviel rückgängig gemacht, so lässt sich mit STRG + Y ein zurückgenommener Schritt wiederherstellen.

Vorsicht: Programmcode in Visual Basic for Applications wird beim Verschieben von Zellen nicht umprogrammiert – dies muss durch den Benutzer manuell durchgeführt werden.

05 Beispiel 2 : Währungsumrechnung

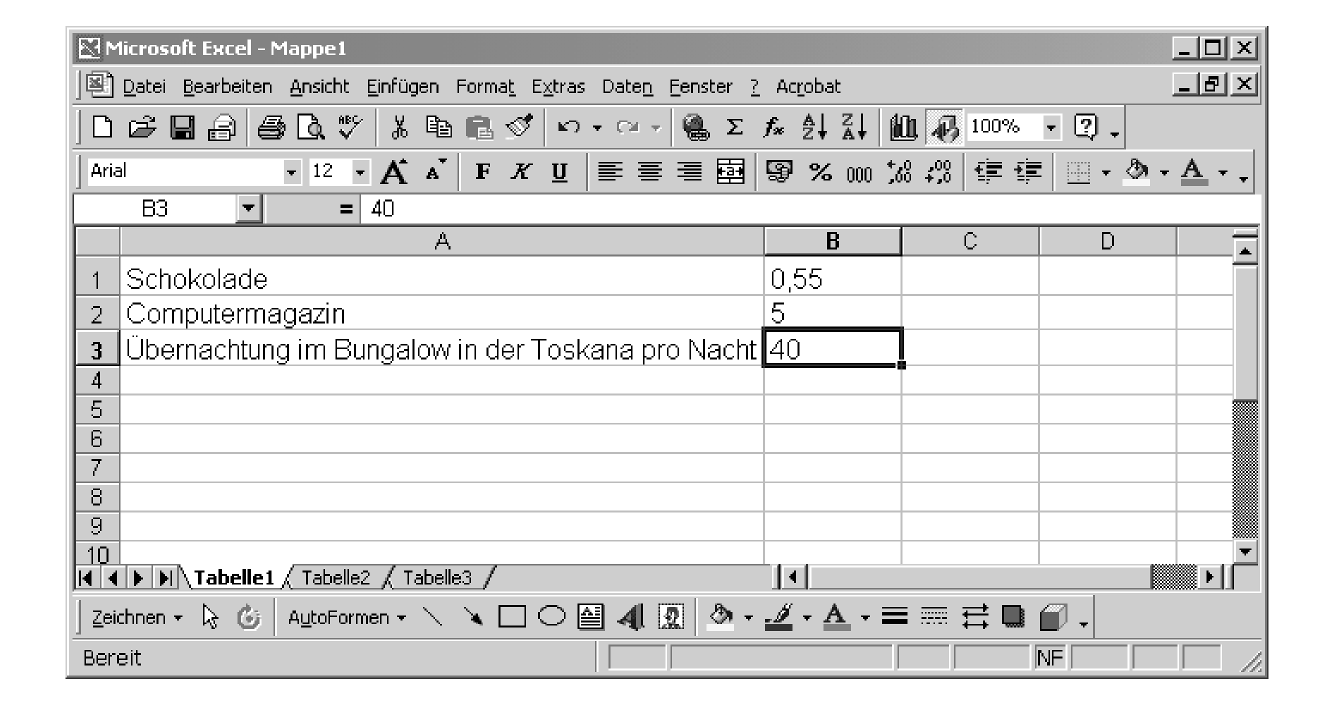
Auch wenn in Europa bedingt durch die Einführung der neuen, gemeinsamen Währung das Umrechen in eine andere Landeswährung quasi aus der Mode gekommen ist, so gibt es doch noch eine Vielzahl, auch potentieller Urlaubsziele, in denen der Euro als Zahlungsmittel nicht akzeptiert wird.

Um auf das Symbol des EURO (€) von Seiten des PC grundsätzlich zugreifen zu können, ist bei Windows 95 oder NT 4.0 und dem Office 97 Paket ein entsprechendes Update (das so genannte Euro- Patch) notwendig. Diesen Patch erhalten Sie im Internet auf den Support- Seiten der Firma Microsoft.

Für das folgende Beispiel sollen folgende Produkte, deren Preis lediglich in EUR bekannt ist in DEM und US$ umgerechnet werden:

Schokolade	0,55 €
Computermagazin	5,00 €
Übernachtung im Bungalow in der Toskana pro Nacht	40,00 €

Sollte sich in der Tabelle auf Ihrem Monitor noch Altdaten aus dem vorherigen Beispiel befinden, so können Sie mit dem Befehl BEARBEITEN -> ALLES MARKIEREN und einem anschließendem Druck auf die ENTF- Taste die Tabelle zurücksetzen. Jedoch wird durch dieses Löschen eventuell vorhandene Formateinstellungen (beispielsweise in Uhrzeit oder Währungseinstellungen).



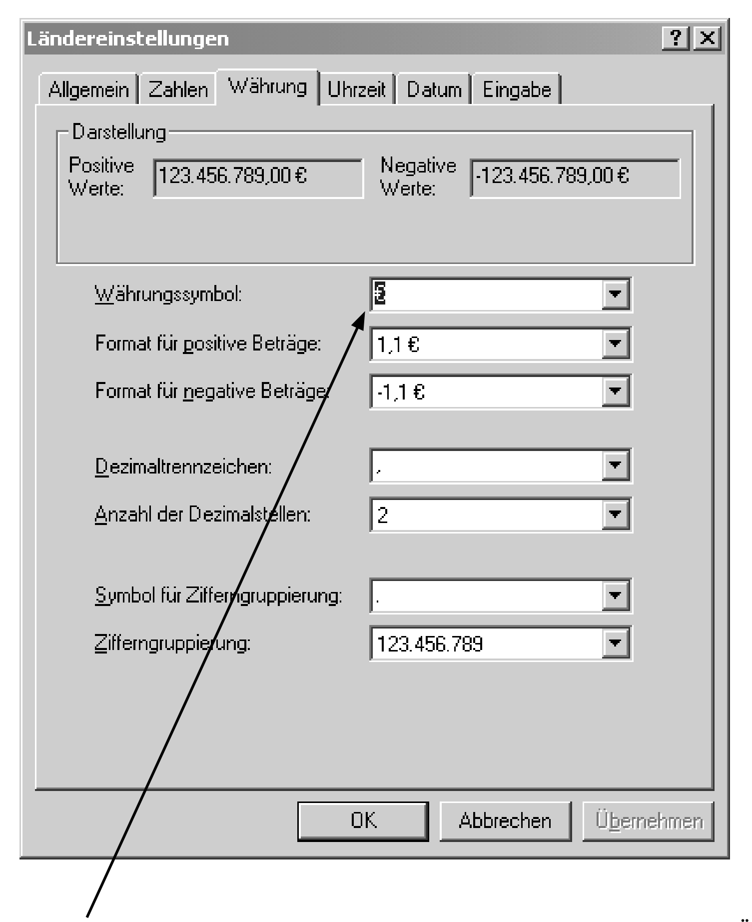
So sollte die Tabelle nach Eingabe der Grunddaten aussehen. Die Werte in der Spalte B sind im Moment noch als Standardzahl formatiert, sprich Microsoft Excel hat keine Kenntnis darüber, dass es sich bei diesen drei Zahlen um eine Währung handelt.

Durch das Markieren des Zellraumes B1 bis B3 (klicken Sie dazu auf B1, halten Sie die linke Maustaste gedrückt und bewegen Sie den Mauszeiger bis zur Position B3) und dem anschließenden Klick auf

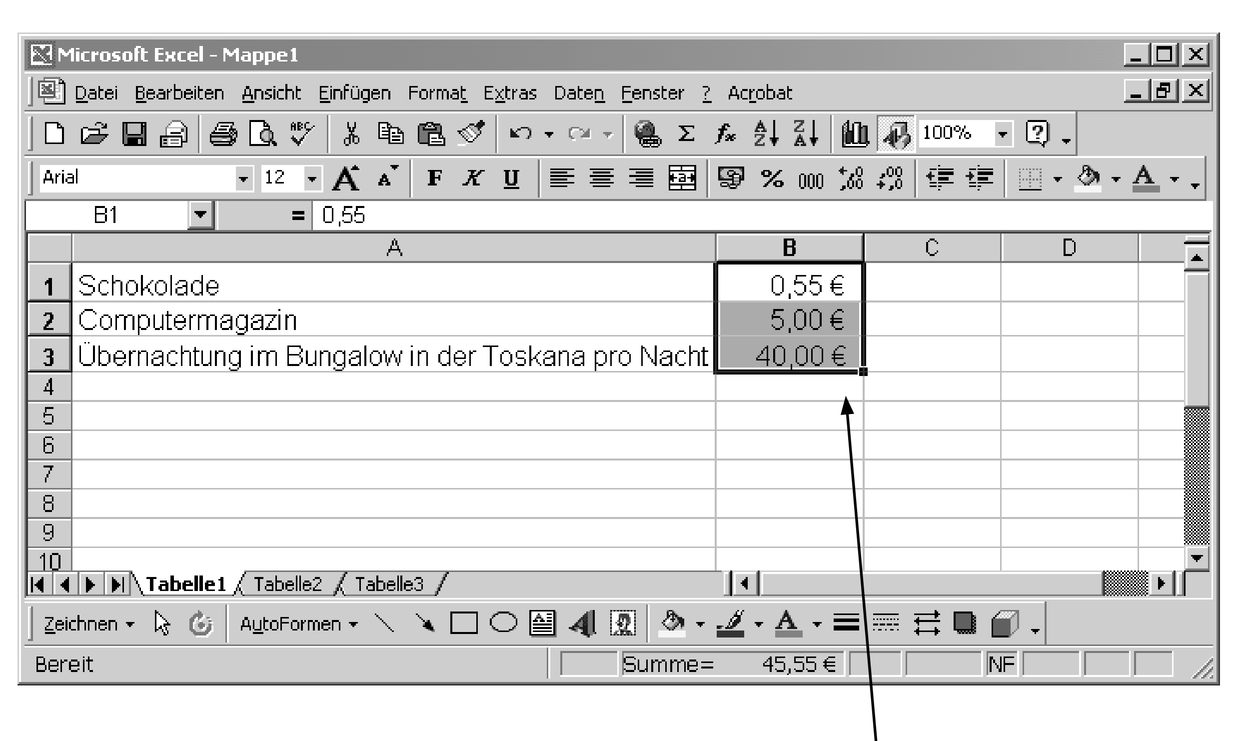
das Währungssymbol (Symbolleiste -> „Schein mit Münzen“) wird der Zellraum auf Währung eingestellt. Sollten Sie direkt das Eurozeichen in der Symbolleiste entdecken, so können Sie auch direkt auf das € - Zeichen klicken.

Mit dem Klick auf das Währungssymbol wird Microsoft Excel veranlasst die Standardwährung, die für Microsoft Windows hinterlegt wurde, auszuwählen.

Erscheint nach der Auswahl anstelle des € **- Zeichens** die Währungsangabe DEM oder DM, so ist Microsoft Excel noch auf den Euro als Standardzahlungsmittel einzustellen. Zu diesem Schritt ist es notwendig die Ländereinstellung in der Systemsteuerung umzustellen (Start -> Einstellung -> Systemsteuerung -> Ländereinstellung -> Register Währung):



Falls der PC anstelle des €- Zeichens die Währung „Deutsche Mark“ angibt ist die Änderung in der Ländereinstellung der Systemsteuerung erforderlich.



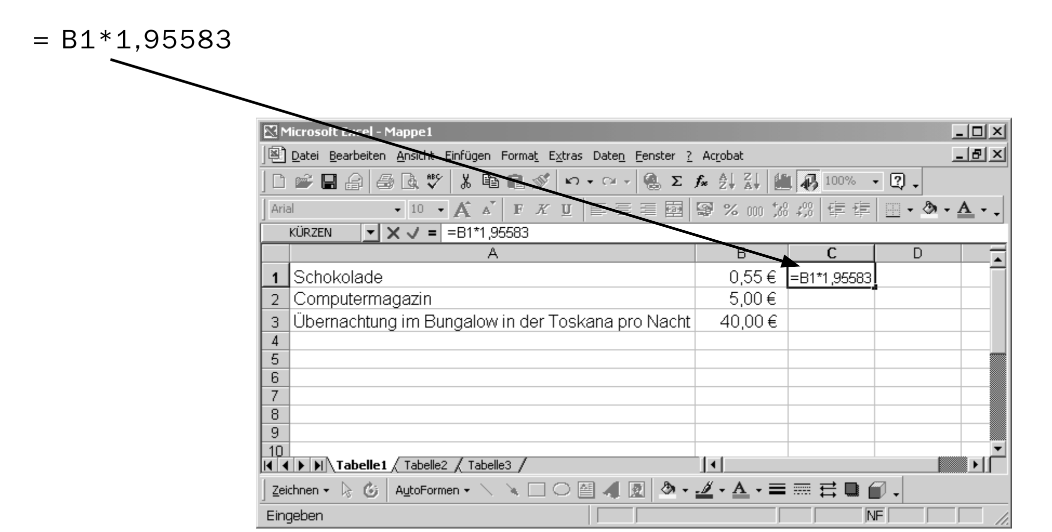
Wenn alle Schritte einwandfrei durchgeführt wurden, so ist jetzt die Währungseinheit in der Beispieltabelle zu erkennen.

Im nächsten Schritt sollen die Euro- Preise in DEM- Preise umgerechnet werden, um beispielsweise einen Preisvergleich mit den Jahren vor der Euro- Einführung durchführen zu können. Das Verhältnis der DEM zum EURO beträgt 1,95583 als offizieller Wechselkurs, sprich für einen Euro mussten 1,95583 DEM gezahlt werden. Somit ist die notwendige Formel:

```
DEM Wert = Eurowert * 1,95583
```

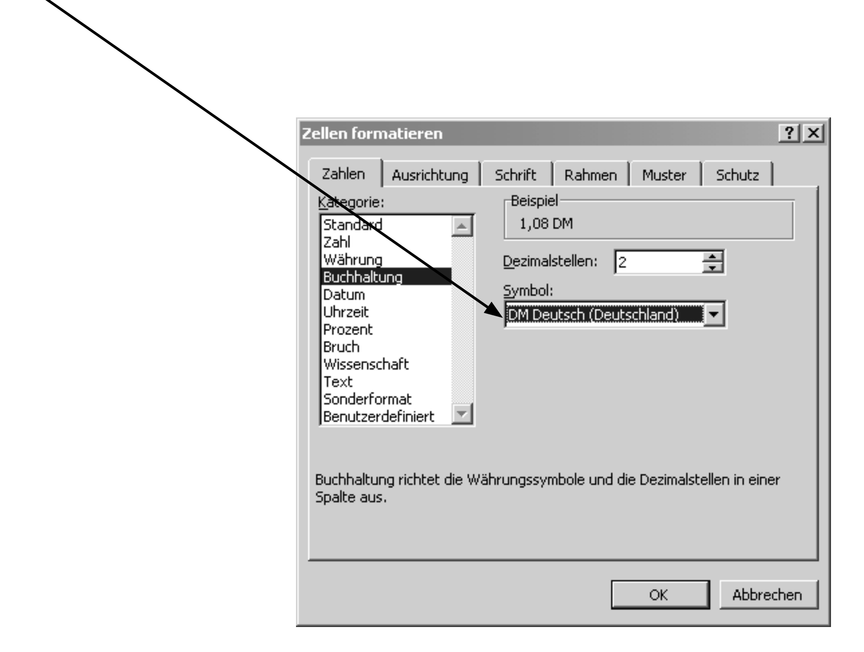
In diesem Beispiel soll in der Spalte C der Preis in DEM angegeben werden, für die Zelle C1 ergibt sich daraus folgende Microsoft Excel Funktion:

```
= B1*1,95583
```



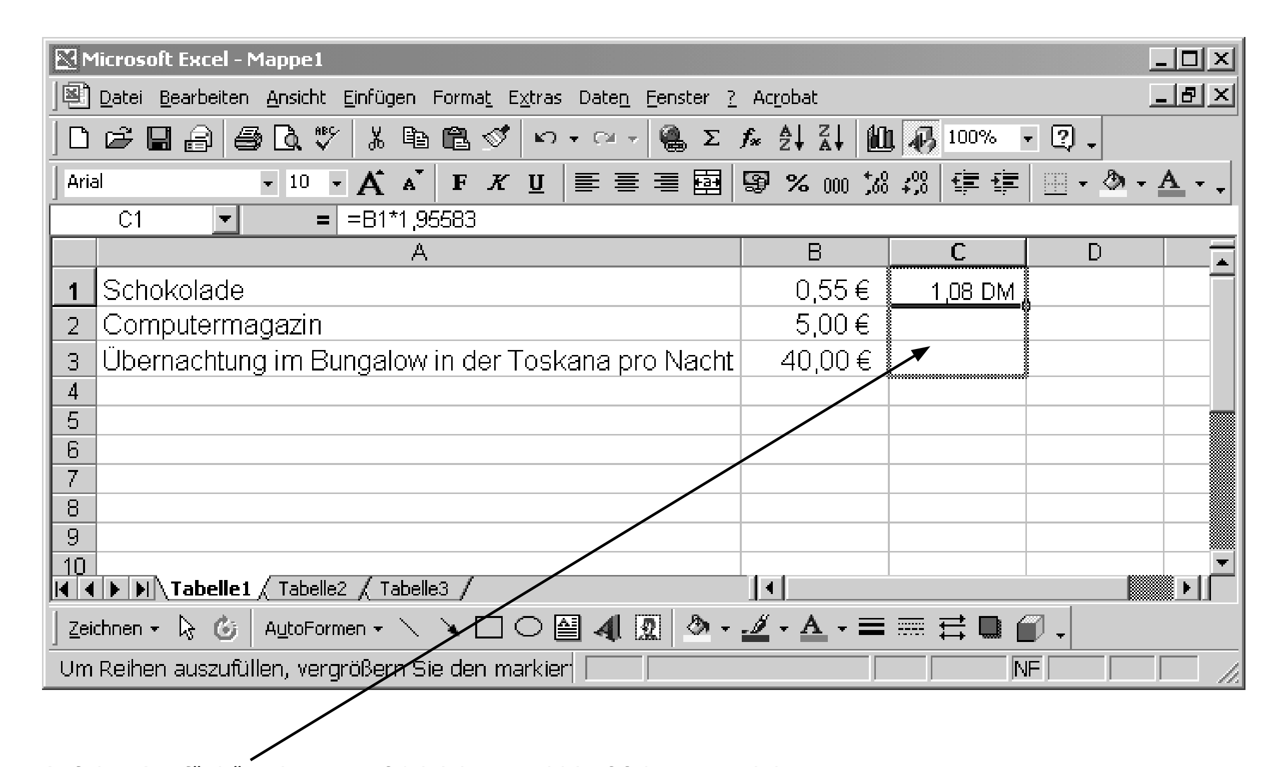
Nach Betätigen der Eingabetaste erscheint das Ergebnis 1,08 € - nun ja, die Zahl stimmt – die Währungseinheit jedoch nicht. Diese muss nun **manuell** auf DEM eingestellt werden. Die Veränderung der Systemsteuerung ist in diesem Fall nicht erforderlich, da ja die Standardwährung nach wie vor EURO lauten soll.

Um die Einstellung dieser Zelle zu verändern ist es notwendig die Zelle C1 zu markieren, in dem der schwarze Auswahlrahmen in der Zelle C1 positioniert ist. Mit einem Klick auf Format -> Zellen... -> Währung lässt sich die bewerkstelligen.



Nach der Umstellung sieht das Tabellenbeispiel wie folgt aus:

Es ist nicht notwendig die Funktion von Hand in die folgenden Zellen einzugeben, dies übernimmt Microsoft Excel automatisch mit Hilfe des Ausfüllkästchens (Abschnitt 5.1).



Auf das Ausfüllkästchen von C1 klicken und bis C3 herunterziehen.

	A	B	C	D
1	Schokolade	0,55 €	1,08 DM	
2	Computermagazin	5,00 €	9,78 DM	
3	Übernachtung im Bungalow in der Toskana pro Nacht	40,00 €	78,23 DM	

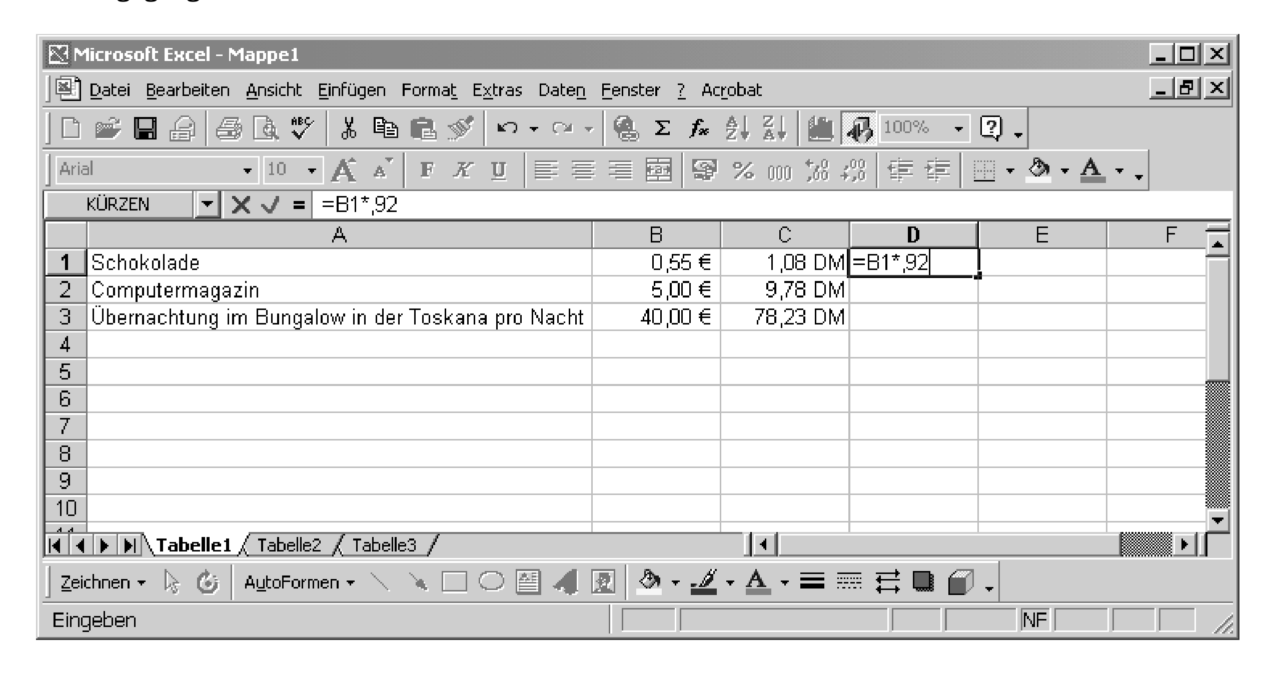
Microsoft Excel hat nun die Funktionen automatisch entsprechend der Bewegungsrichtung der Mausbewegung eingetragen. Für Zelle C2 ist dies:

= B2 * 1,95583

und für die Zelle C3 :

= B3 * 1,95583

Um nun die Währungsumrechnung in US$ durchzuführen könnte nach dem oben beschriebenen Verfahren vorgegangen werden:



Anschließend ist das Format auf „Währung US$“ umzustellen.

Diese Vorgehensweise hat jedoch einen entscheidenden Nachteil: Ändert sich das Verhältnis zwischen Euro und US- Dollar, so ist der Benutzer gezwungen die Funktionen jedes Mal manuell zu ändern und mit Hilfe des Ausfüllzeichens aufzufüllen.

05.1 Der Zelle einen Namen geben

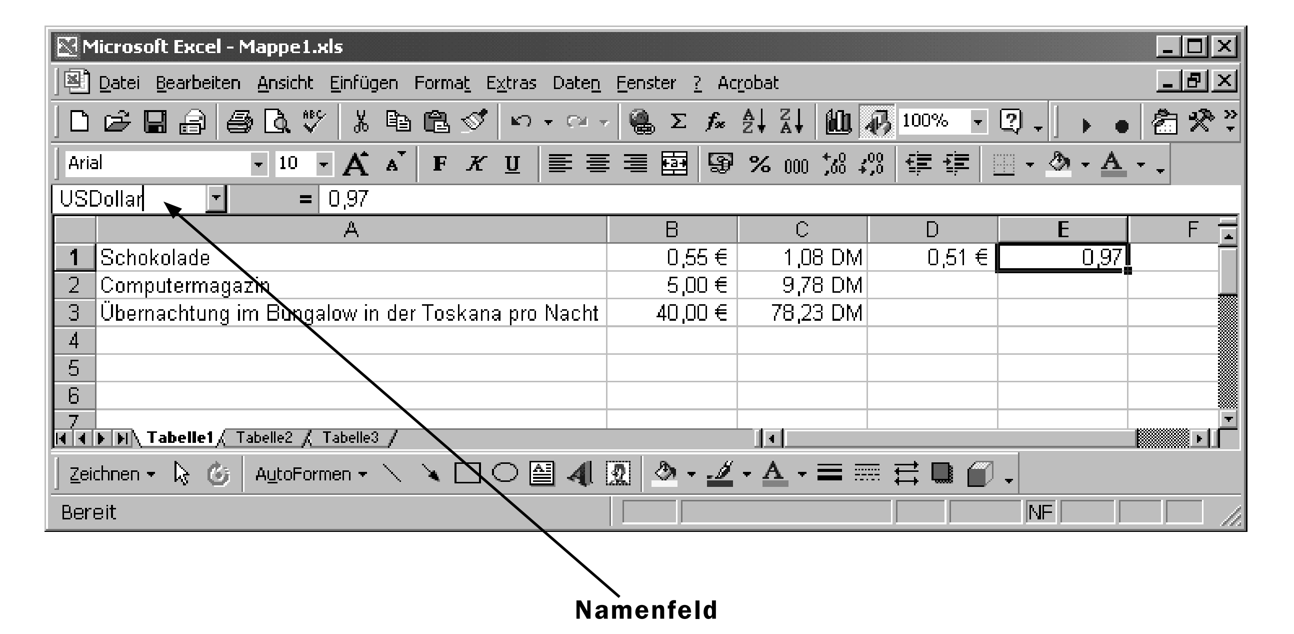
Praktischer wäre es an einer Stelle der Tabelle den Währungskurs Euro – US$ abzulegen und nur den Verhältniseintrag (sprich: den aktuellen US$- Kurs zu ändern). Um dies zu bewerkstelligen ist eine neue Methode einzusetzen, die Vergabe von **Zellnamen**.

In Zelle E1 wird der aktuelle Dollarwert beispielsweise 0,97 eingegeben.

Nun soll diese Zelle E1 nicht nur unter dem Namen E1 innerhalb des Tabellenverbundes bekannt sein, sondern auch unter dem Namen „USDollar“. Dieser Name wird nun in das Namenfeld eingegeben, oder über

EINFÜGEN -> NAMEN -> DEFINIEREN

festgelegt. Bestätigen Sie die Eingabe in jedem Fall mit der Eingabetaste.



Die so festgelegten Bezeichnungen („Namen“) müssen innerhalb der gesamten Arbeitsmappe eindeutig sein, sprich es kann keine doppelten Namen geben. Ist der Name festgelegt, so kann dieser in jeder Tabelle der Arbeitsmappe in Formel verwendet werden, beispielsweise

= USDollar

oder

= 4 * USDollar/2.

In dem hier beschriebenen Beispiel der Währungsumrechnung ändert sich somit die Funktion in Zelle D1 zu = B1 * USDollar

Jetzt lässt sich die Funktion mit dem Ausfüllkästchen „herunterziehen“ und die Berechnungen jederzeit durch Veränderung des „Euro: US$“- Verhältnisses bequem ändern.

05.2 Feste Verweise auf Zellen

Nicht immer ist die Vergabe von Zellnamen, quasi als Variablennamen, eine sinnvolle Vorgehensweise. Unter Microsoft Excel gibt es noch eine weitere Möglichkeit einen Verweis auf eine Zelle „festzuhalten“ und zwar durch den Einsatz des Dollar- Zeichens. Weitere Information zum Ausfüllkästchen finden sich im Abschnitt 5.3.



Um dafür zu sorgen, dass beim „Auffüllen“ einer Funktion der Verweis auf die Zelle E1, die in unserem Beispiel den US- Dollar Kurs repräsentiert nicht auf E2, E3 und so weiter verändert wird, kann die Formel in D1 wie folgt verändert werden:

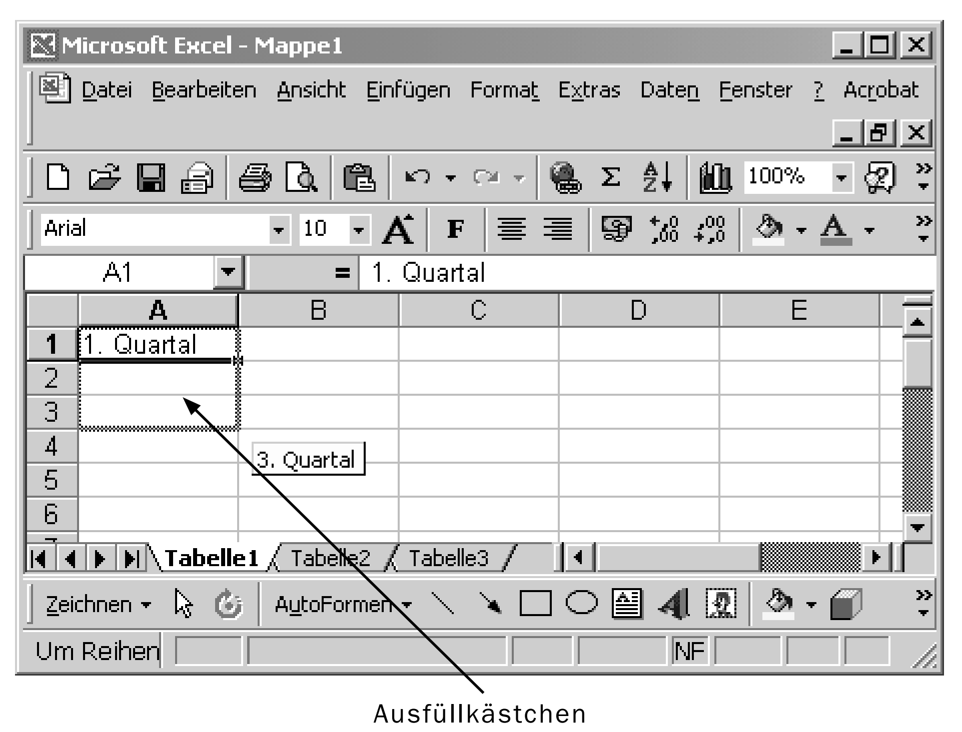
```
=B1*$E$1
```

Das Dollarsymbol vor der Positionsangabe E und 1 hält diesen Verweis fest.

Nun kann die Funktion am Ausfüllkästchen ohne Probleme „heruntergezogen“ werden, nur der erste Teil der Funktion [= B1] wird entsprechend auf B2 und B3 korrigiert, der Verweis auf E1 bleibt durch die Dollarzeichen stabil.

05.3 Ausfüllen von Daten

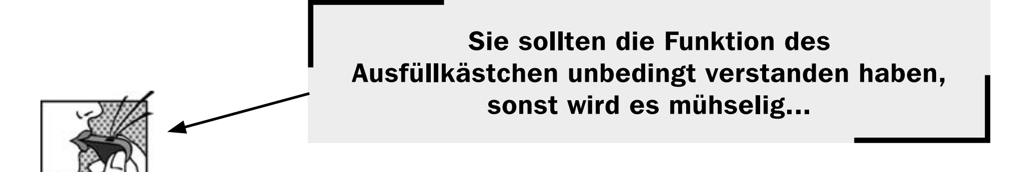
Zellen können mit mehreren Typen von Datenreihen ausgefüllt werden, indem Sie die gewünschten Zellen markieren und dann das **Ausfüllkästchen** ziehen oder den Befehl Reihe verwenden (zeigen Sie im Menü Bearbeiten auf Fläche, und klicken Sie dann auf Reihe).



Kopieren von Daten innerhalb einer Zeile oder Spalte
Indem Sie das Ausfüllkästchen einer Zelle ziehen, können Sie den Inhalt einer Zelle in andere Zellen in der gleichen Zeile oder Spalte kopieren.

Ausfüllen einer Reihe von Zahlen, Datumswerten oder anderen Elementen
Wenn eine Auswahl eine Zahl, ein Datum oder einen Zeitraum enthält, können Sie die Reihe erweitern. Beispielsweise ergibt sich aus der Anfangsauswahl der Uhrzeit in der folgenden Tabelle als Ergebnis die gezeigte Reihe. Elemente, die durch Kommas getrennt sind, befinden sich in angrenzenden Zellen.

Anfangswert	Erweiterte Reihen
1, 2, 3	4, 5, 6
9:00	10:00, 11:00, 12:00
Mo	Di, Mi, Do
Montag	Dienstag, Mittwoch, Donnerstag
Jan	Feb, Mrz, Apr
Jan, Apr	Jul, Okt, Jan
Jan 99, Apr 99	Jul 99, Okt 99, Jan 00
15. Jan, 15. Apr	15. Jul, 15. Okt
1999, 2000	2001, 2002, 2003
1. Jan, 1. Mrz	1. Mai, 1. Jul, 1. Sep...
Qrtl3 (bzw. Q3 oder Quartal3)	Qrtl4, Qrtl1, Qrtl2...
Produkt 1: Nicht erledigt	Produkt 2: Nicht erledigt, Produkt 3: Nicht erledigt...
Text1, TextA	Text2, TextA, Text3, TextA...
1. Periode	2. Periode, 3. Periode...
Produkt 1	Produkt 2, Produkt 3...



06 Analysen und Grafiken mit Excel

Eine der großen Stärken moderner Tabellenkalkulationsprogramme liegt in der unmittelbaren Visualisierung durch die Erstellung von Grafiken aufgrund des Datenmaterials.

Um beispielsweise einen **Trend** (also die erwartete Fortführung von Daten) zu erstellen, wird eine Funktion verwendet, die bereits im Kapitel 5.3 (Auffüllen von Daten) erörtert wurde. Mathematisch gesehen ist der lineare Trend eine Berechnung auf Methode der kleinste Quadrate.

Für den Leser, die Leserin ist das Auffüllen der Reihe :

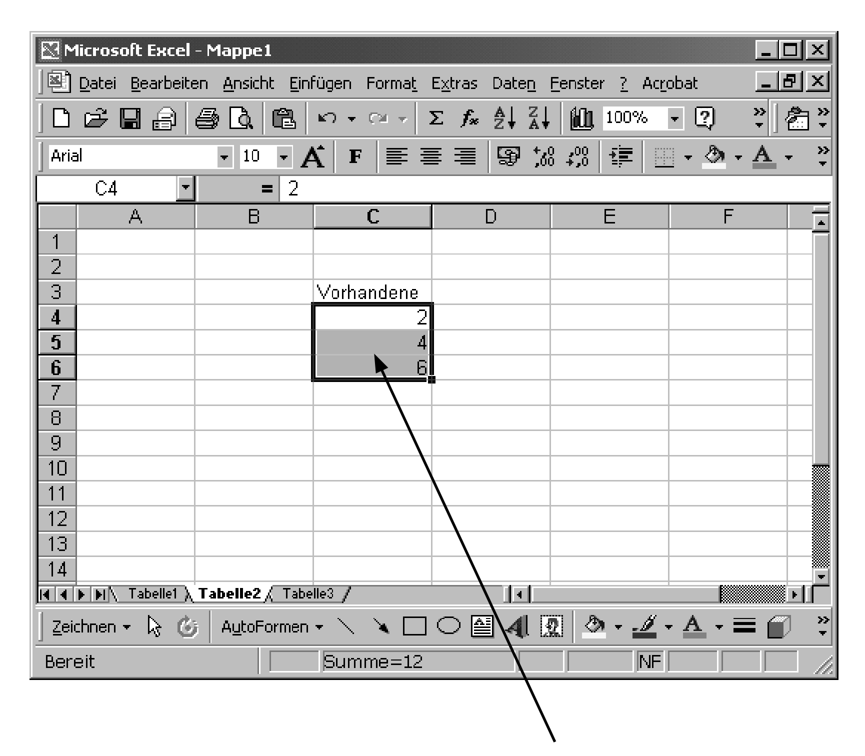
2,4,6,8 (bekannte Zahlen) 10,12,14,16 (gedankliche Fortführung)

oder

2,4,16,256 (bekannte Zahlen) 65536, 281474976710656 (Fortführung)

ein einfaches (mehr oder weniger) unterfangen.

Eintragen der Werte in Microsoft Excel :

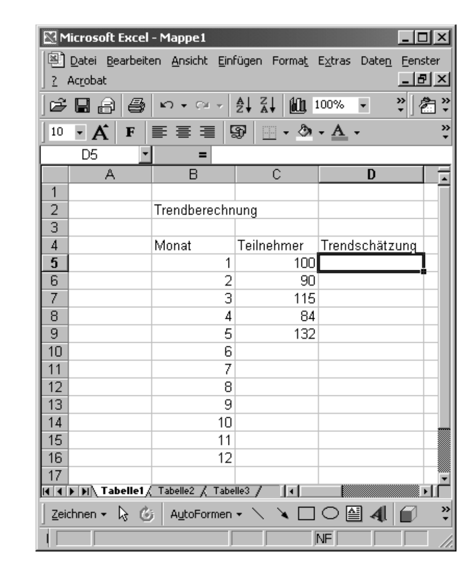


Markierung der vorhandenen Zahlen und bewegen des **Ausfüllkästchen** sorgt für eine Fortführung der Reihe.

Wird jedoch der Trend für eine „Was wäre wenn" Berechnung benötigt, so ist es sinnvoll die Funktion direkt zu verwenden. Aufgrund der Matrix (mehrere Ergebnisebenen) ist die Funktion komplexer, als bisher kennen gelernte Funktionen.

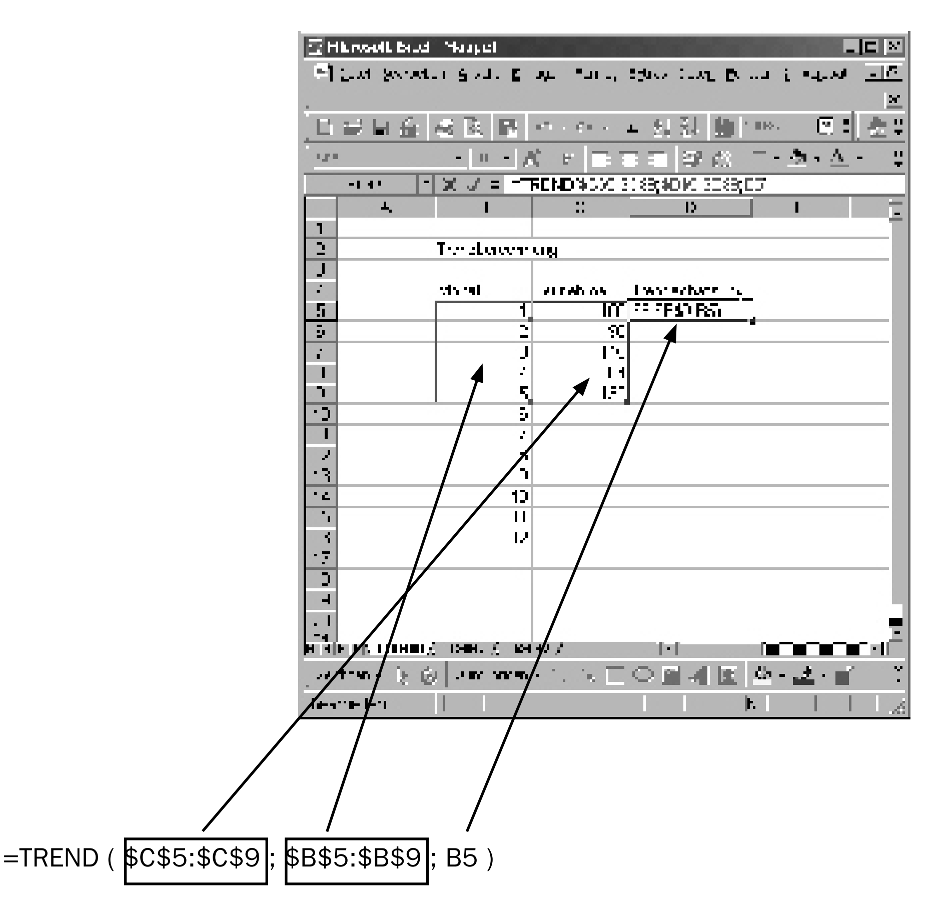
Die Trendfunktion ist wie folgt aufgebaut

= TREND(<bekannte Y- Werte>; <bekannte X- Werte>; <Neue X- Werte>;Konstante)



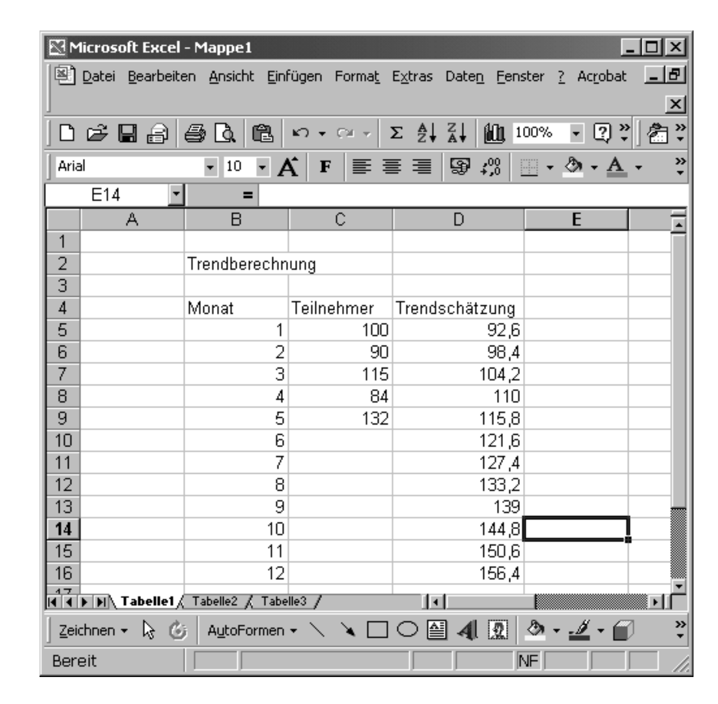
Die Tabelle aus obiger Abbildung soll als Lehrbeispiel für die **Trendberechnung** dienen, eine Trendanalyse für Teilnehmer für die Monate Juni bis Dezember. Bekannt sind lediglich die Teilnehmer aus den Monaten Januar bis Mai. Da Microsoft Excel die Monate für die Berechnung benötigt, kann nicht der Monatname direkt in die Berechnungsspalte eingegeben werden.

Um Microsoft Excel davon abzuhalten die Formel stets beim „Runterziehen" umzuschreiben wird mit dem Dollarzeichen ein Umschreiben der Formel verhindert. Die Trendformel für D5 lautet :



=TREND ($C$5:$C$9 ; $B$5:$B$9 ; B5)

Das Ergebnis der Funktion ist 92,6, wird diese Formel mit dem Ausfüllkästchen „heruntergezogen“ bis auf D16 so ergibt sich folgende Trendberechung für die Entwicklung von Teilnehmern im Beispielsgeschäftsjahr :



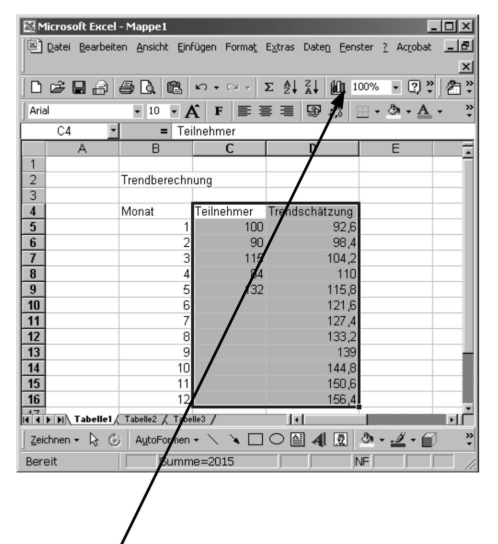
Trendberechnung

Monat	Teilnehmer	Trendschätzung
1	100	92,6
2	90	98,4
3	115	104,2
4	84	110
5	132	115,8
6		121,6
7		127,4
8		133,2
9		139
10		144,8
11		150,6
12		156,4

Durch Änderung der Teilnehmerzahlen aus den Monaten Januar bis Mai kann nun die Abhängigkeit des Trends von einzelnen Werten der Daten begutachtet werden. Die **Trendschätzung** der Monate, in denen Realdaten vorliegen zeigt wiederum wie weit sich Schätzung und tatsächliches Ergebnis voneinander unterscheiden können.

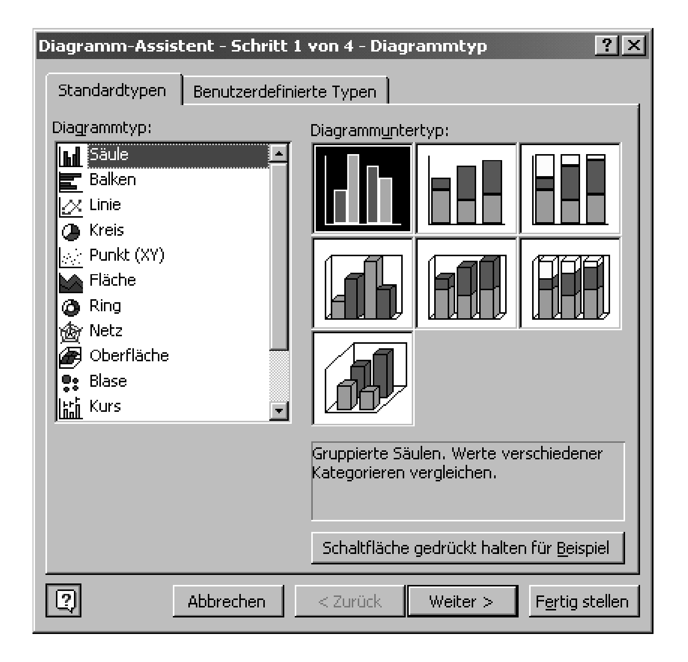
Gemäß dem Volksmund sagt ein Bild mehr als tausend Worte. Dieses Trendabbild soll nun in eine Grafik verwandelt werden.

Um dies zu bewerkstelligen ist der Bereich, der für die Grafik relevant sein könnte zunächst zu markieren. In unserem Fall wäre das der Bereich C4 bis D16.

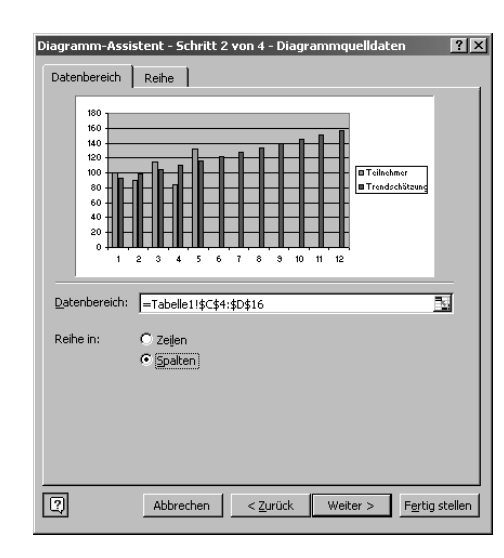


Anschließend ein Klick auf das Diagrammsymbol oder auf die Befehle EINFÜGEN -> DIAGRAMM...

Direkt nach dem Klick erscheint ein **Diagramm- Assistent** der in vier Schritten die Erstellung von Diagrammen unterstützt. Im ersten Schritt ist in der linken Fensterhälfte der Typ (Säule, Balken, Linie...) auszuwählen und in der rechten Fensterhälfte der Untertyp. Im unteren, rechten Fenster- Viertel erscheint eine Kurzbeschreibung über den ausgewählten Diagrammtyp. Mit „Schaltfläche gedrückt halten für Beispiel“ lässt sich das bisher gewählte Diagramme in eine Vorschau betrachten.

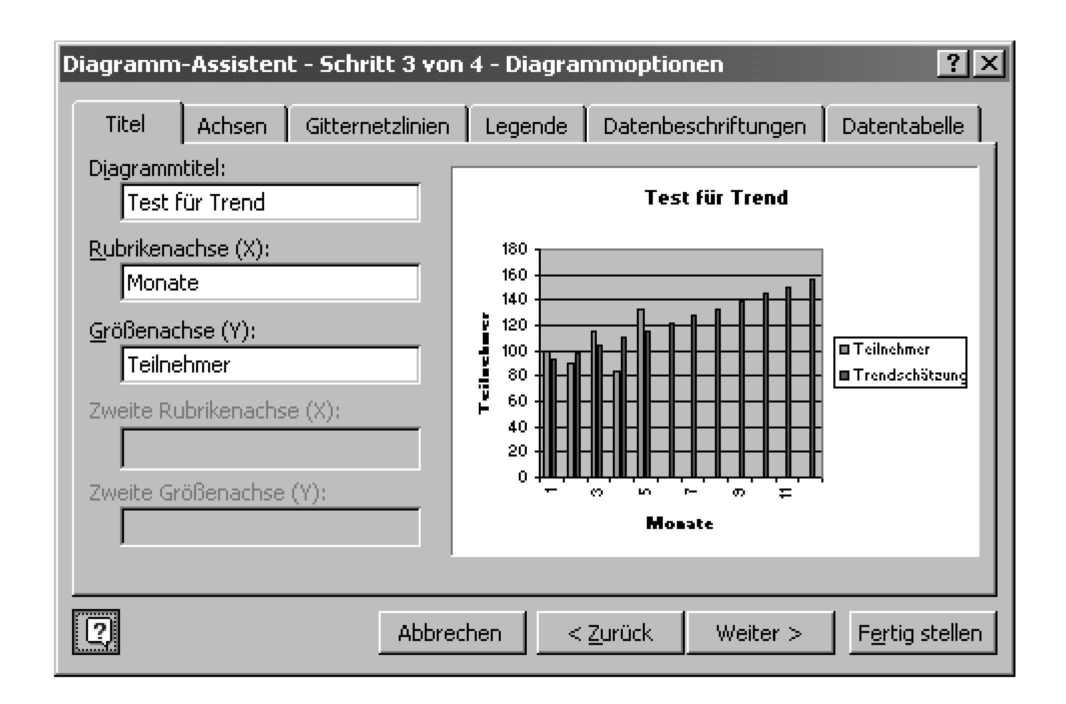


Für das Trendbeispiel klicken Sie einfach auf „Weiter >“.



Im zweiten Schritt ist festzulegen in „welcher Richtung“ Microsoft Excel das Datenmaterial durcharbeiten soll, um das Diagramm zu erstellen. Um die Veränderung zu beobachten, kann das Optionsfeld „Reihe in o Zeilen“ angewählt werden, die Vorschau produziert umgehend ein Vorab- Ergebnis.

Nach einem Klick auf „Weiter >“ wird das Diagramm um weitere Informationen erweitert :



Die Register Achsen, Gitternetzlinien, Legende, Datenbeschriftung und Datentabelle legen weitere Lay- out- Eigenschaft für das Diagramm fest. Anschließend mit „Weiter >“ zum letzten Assistenten- Schritt :

Diagramm-Assistent - Schritt 4 von 4 - Diagrammplatzierung

Diagramm einfügen:

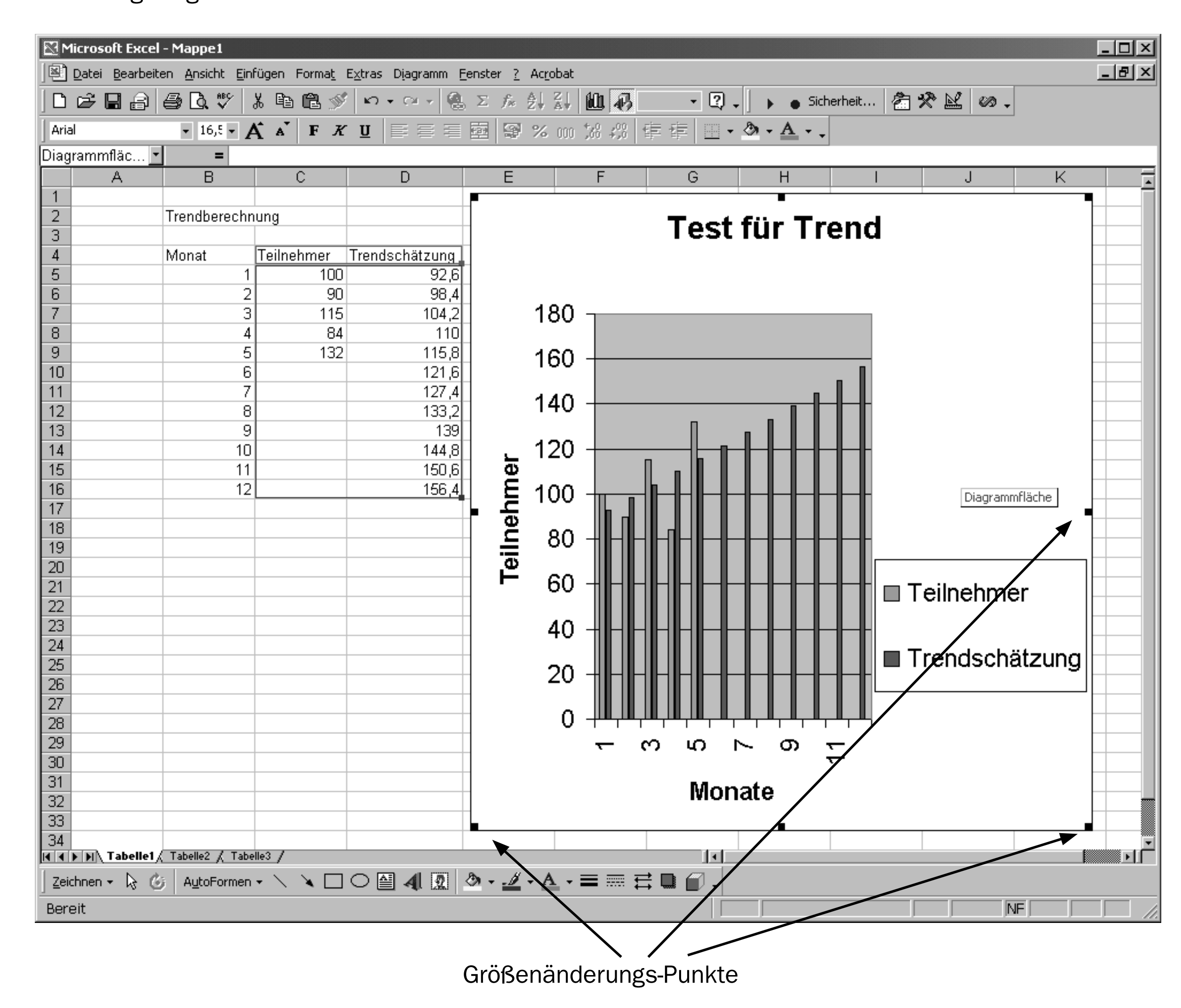
Als neues Blatt: Diagramm1

Als Objekt in: Tabelle1

Abbrechen | < Zurück | Weiter | Fertig stellen

Wird „Als Objekt in...“ ausgewählt, so erscheint das Diagramm als frei bewegliches Diagramm in der aus- gewählten Tabelle. Klick auf „Fertig stellen“.

Das fertige Ergebnis :



Wird nun ein Wert in der Datentabelle verändert, so wird das Diagramm automatisch aktualisiert. Die Größe des Diagramms kann an den acht Objekt- Größenänderungs-Punkten verändert werden.

Auch innerhalb des Diagrammfensters lassen sich Elemente nach wie vor verändern.

Um alle Funktionalitäten des Diagramm- Assistenten überblicken zu können empfiehlt es sich „herum- zuspielen“.

06.1 Diagramm an MS Word

Häufig werden die so generierten Diagramme in Word- Dateien benötigt, um die dortigen, „textlastigen“ Informationen aufzulockern.

Es werden drei Methoden der Weitergabe unterschieden ->

- die Übergabe der Grafik als Bild, ohne die Möglichkeit der Veränderung
- die Übergabe der Grafik mit hinterlegtem Datenmaterial, ohne automatische Aktualisierung
- die Übergabe der Grafik mit verknüpftem Datenmaterial aus der Datei

Die Weitergabe als Bild hat zwei Vorteile, wobei der eine Vorteil je nach Ziel auch ein Nachteil darstellt. Bilder benötigen zwar einigen Speicherplatz, lassen sich aber durch Kompressionsroutinen wie dem ZIP-Algorithmus (in Windows XP bereits in das Betriebssystem integriert, unter anderen Windows- Versionen durch Programme wie PowerArchiver oder WinZIP verfügbar) gut „schrumpfen“.

Ein Bild kann vom späteren Empfänger nicht ohne enorme Mehrarbeit verändern, sprich die so weitergegebene Word- Datei ist **komplizierter zu manipulieren**. Da das Bild unabhängig von der ursprünglichen Anwendung (in diesem Falle Microsoft Excel ist), benötigt der Empfänger der Datei nicht zwangsläufig selbst Microsoft Excel auf seinem Computer.

Um das Diagramm als Bild in Microsoft Word einzufügen gehen Sie wie folgt vor :

1. Markieren des Diagramm in Microsoft Excel
2. Rechtsklick
3. „Kopieren" im Kontextmenü auswählen
4. Wechsel zu Microsoft Word
5. Menübefehl „Bearbeiten"
6. „Inhalte Einfügen als..."
7. -> Bild (Erweiterte Metadatei)
8. Größe lässt sich an den acht Punkten ändern

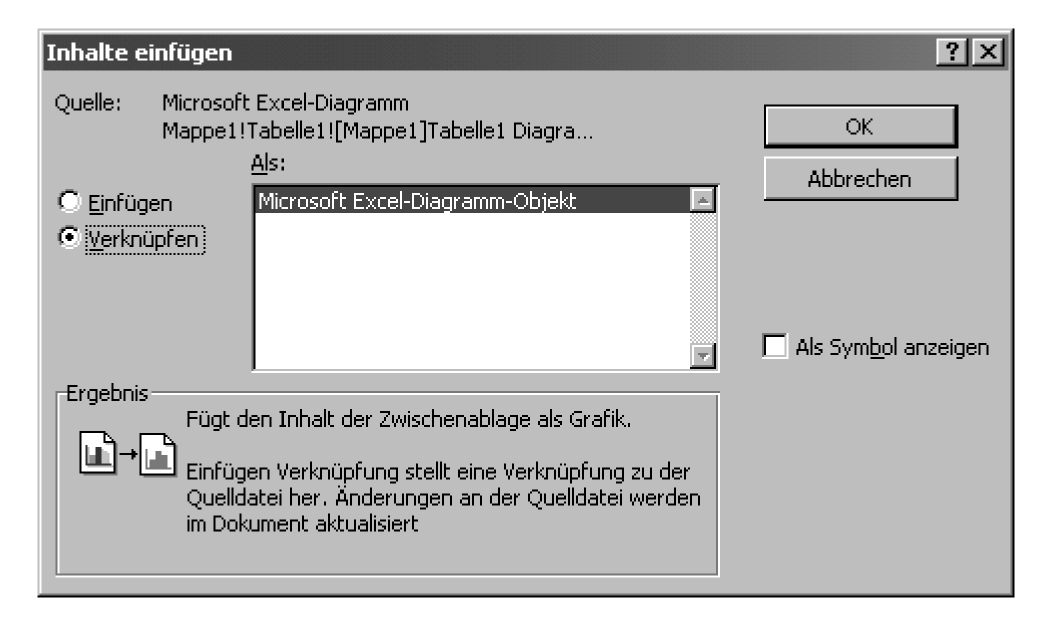
Die Weitergabe des Diagramm mit hinterlegtem Datenmaterial hat den Vorteil das die Microsoft Excel-Tabelle mit in das Microsoft Word- Dokument übergeben wird und durch einen Doppelklick auf das Diagramm wieder zum Vorschein gebracht werden kann. Das Diagramm bleibt somit **ÄNDERBAR**!

1. Markieren des Diagramm in Microsoft Excel
2. Rechtsklick
3. „Kopieren" im Kontextmenü auswählen
4. Wechsel zu Microsoft Word
5. Menübefehl „Bearbeiten"
6. Einfügen

Die dritte Methode verbindet die Microsoft Word- Datei direkt mit dem Datenmaterial in der Microsoft Excel- Datei. **Somit wirken sich Änderungen der Microsoft Excel- Datei beim nächsten Öffnen der Microsoft Word- Datei direkt auf das dort abgelegte Diagramm aus**. Diese Methode ist besonders in Netzwerk- Umgebungen oder bei sich häufig ändernden Dokumenten praktisch.

Wird jedoch die ursprüngliche Datei nicht mehr gefunden, so verbleibt das Microsoft Word- Dokument im vorherigen Zustand. Die Aktualität des Diagramms ist somit etwas schwieriger zu überschauen!

1. Markieren des Diagramm in Microsoft Excel
2. Rechtsklick
3. „Kopieren" im Kontextmenü auswählen
4. Wechsel zu Microsoft Word
5. Menübefehl „Bearbeiten"
6. „Inhalte einfügen..."
7. in linker Fensterhälfte „Verknüpfen" auswählen



Die Quelle bezeichnet nun die Ursprungsdatei.

07 Logikoperatoren: Tue dies wenn das...

Sicherlich werden Sie es als Leser bereits bemerkt haben, mit Microsoft Excel gerät man bereits auf der Tabellenebene in die Nähe der Arbeit eines **Programmierers**. Zunächst scheinen viele Funktionen fremd und ungewöhnlich, um nichts das Wort „kompliziert" zu benutzen. Doch der Einsatz von Microsoft Excel ist lohnenswert, in keinem anderen Programm haben Sie die Möglichkeit Informationen in dieser Geschwindigkeit zu bearbeiten.

Natürlich ist dem Einsatz von Microsoft Excel eine **Grenze** gesetzt, ab einer Anzahl von mehr als 50.000 Informationssätzen ist der Einsatz eines Datenbanksystems wie Delphi oder Microsoft Access sicherlich sinnvoller.

In diesem Abschnitt geht es um folgende Funktionalität: Microsoft Excel soll in Abhängigkeit von einem Wert entweder dies oder etwas anderes tun. Somit ist die Entscheidung, was weiter getan wird abhängig von einem logischen Operator. Stellvertretend für die Gruppe der Logikbefehle soll als erstes der Befehl NICHT betrachtet werden.

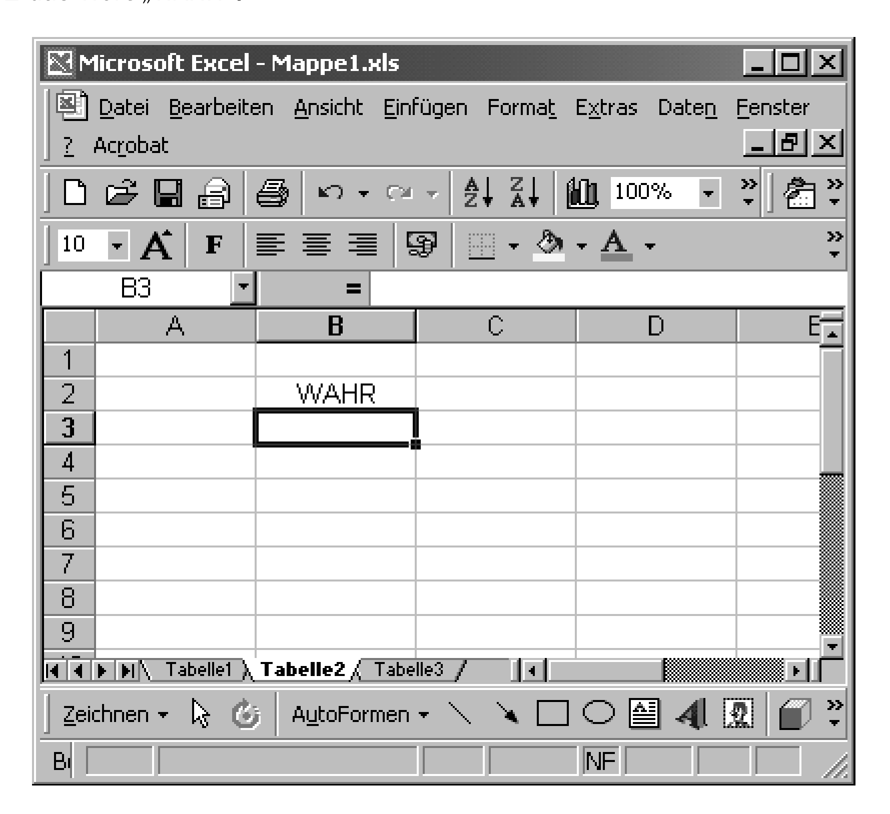
Gemäß der Beschreibung von Microsoft Excel kehrt der Befehl NICHT den Wahrheitswert um.

Für die folgenden Beispiele soll eine neue Tabelle verwendet werden, schließen Sie eventuell vorhandene Tabellen mit dem Befehl DATEI -> SCHLIESSEN.

Tippen Sie bitte in die Zelle B2 folgende Funktion:

= NICHT(B1>5)

Übersetzt lautet der Befehl (B1 ist NICHT größer als 5). Sobald Sie die Eingabetaste gedrückt haben, erscheint in B2 das Wort „WAHR".



Warum „WAHR" ?

Weil das **Ergebnis** richtig ist, denn B1 ist nicht größer als 5, sondern die Zelle B1 hat gar keinen Wert (aber auch nicht 0 – sondern NICHTS).

Wird nun in B1 0,1,2,3,4 oder 5 eingegeben so verändert sich erwartungsgemäß das Ergebnis in B2 nicht, es bleibt weiter „WAHR“. Erst wenn eine Zahl größer oder gleich sechs eingegeben wird erscheint als Ergebnis in B2 „FALSCH“.

Nun liegt die Frage nahe, was soll ich denn mit den Worten „FALSCH“ und „WAHR“ anfangen? Kann ich denn mit solchen Ergebnissen weiterrechnen? Ja, das ist Möglich – den das **boolesche Algebra**, basierend auf den Aussagen 0 (WAHR) und 1 (FALSCH) ist die Grundlage für die logischen Operatoren und nicht nur das. Im Kern Ihres Computers arbeitet eine so genannte CPU (Central Processing Unit = Hauptprozessor mit, raten wir mal, 500 MHz Taktfrequenz). Das bedeutet schlicht und etwas ungenau erklärt, Ihr Hauptprozessor vollzieht 500 Millionen Operationen in der Sekunde die alle genau auf 0 und 1 (dem binären System) basieren. Ein Computer kann genau genommen nicht mehr als 0 und 1 von einander zu unterscheiden, nur das unheimlich schnell – das war jedoch nur eine Randnotiz.

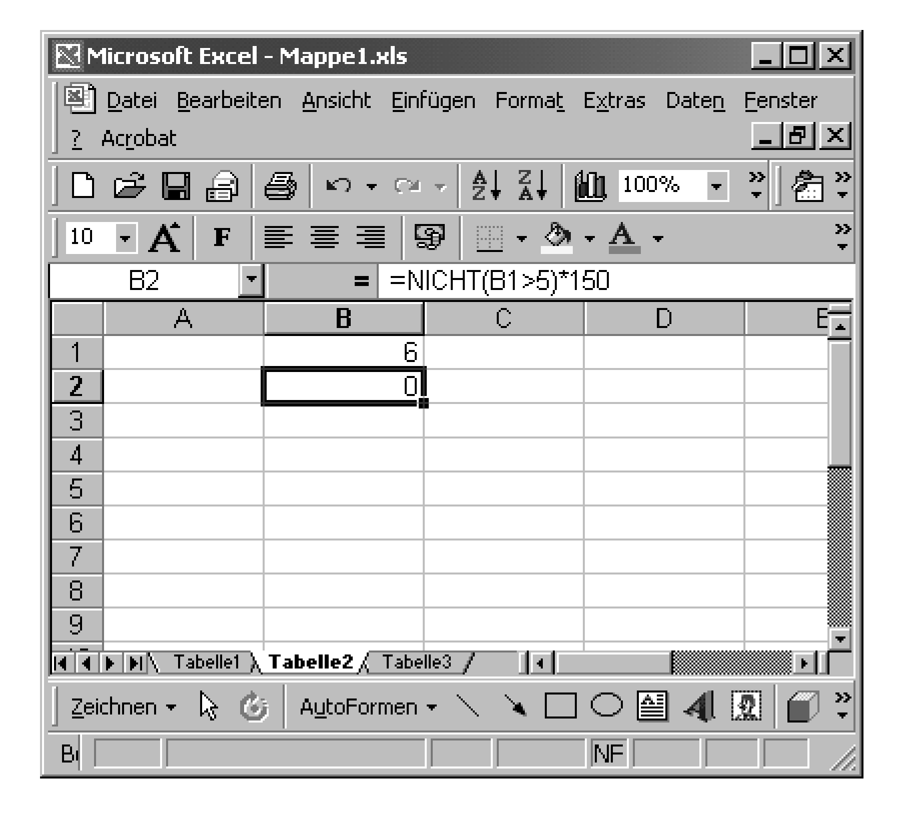
Mit Null und Eins (repräsentiert durch FALSCH und WAHR) lässt sich ganz elegant weiterrechnen. Stellen Sie sich vor, die soeben erstellte Tabelle soll einen tariflichen Zuschlag für den Nachtdienst berechnen. Erst wenn mehr als 5 Nächte innerhalb eines Monats Dienst absolviert wird, soll ein Zuschlag von € 150 gewährt werden.

Wir werden in dem Beispiel nun nicht die Kalkulation der Dienstformen durchführen, dass soll zu einem späteren Zeitpunkt im Script betrachtet werden – hier sei nur die logische Funktion von Interesse.

Wird nun die Formel in B2 entsprechend verändert :

= NICHT(B1>5)* 150

Erscheint, sofern eine Zahl größer oder gleich sechs in B1 zu finden ist, jedoch als Ergebnis 0.



Dies erklärt sich durch die im Befehl NICHT durchgeführte Umkehrung der Funktion. Denn 1 * 150 ist bekanntlich 150, aber 0 * 150 ist null. Somit muss in diesem Beispiel die Funktion erneut verändert werden :

= NICHT(B1<=5)*150

Somit wird die Zulage von €150 gewährt, sobald die Anzahl von Diensten in B1 größer oder gleich fünf ist.

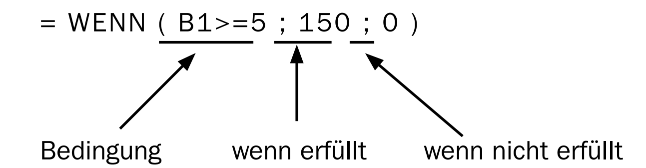
07.1 WENN

Ein ganz wichtiger Befehl aus der Gruppe der Logikbefehle ist der Befehl WENN, der aus insgesamt **drei Teilen** besteht.

Im ersten Teil des Befehls wird eine Bedingung festgelegt, in unserem Beispiel „B1>=5“, also mehr oder gleich fünf Dienste. Im zweiten Teil wird festgelegt, was Microsoft Excel tun soll wenn die Bedingung des ersten Teils erfüllt wird und der dritte Teil legt fest, was zu tun ist, wenn die Bedingung nicht erfüllt wird.

Tippen Sie bitte folgende Funktion in Zelle B2 ein :

= WENN (B1>=5 ; 150 ; 0)

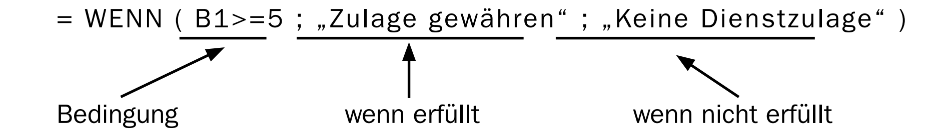


Zwischen den Befehlsteilen muss ein Semikolon stehen !

Somit erfüllt der WENN- Befehl die gleiche Funktion wie der zuvor betrachtete NICHT- Befehl. Jedoch ist es möglich mit dem WENN Befehl auch ganze Textausgaben zu produzieren, die dem Tabelleanwender Informationen geben.

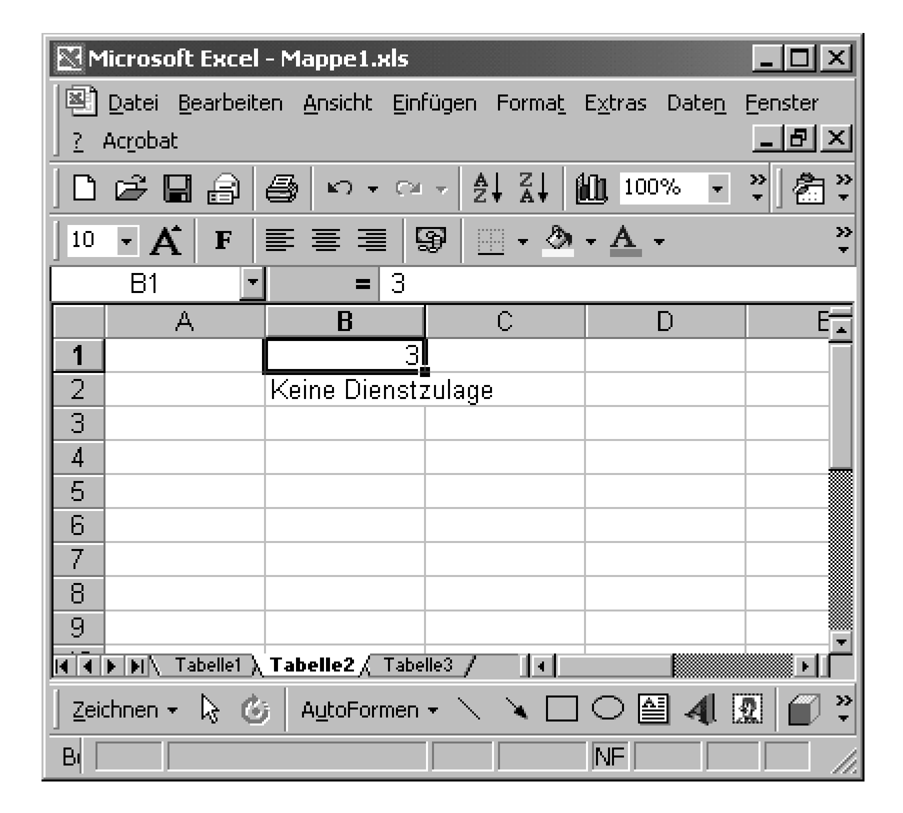
Dazu muss die Funktion in B2 wie folgt verändert werden :

= WENN (B1>=5 ; „Zulage gewähren“ ; „Keine Dienstzulage“)



Durch die Anführungszeichen in dieser Funktion wird Microsoft Excel dazu angewiesen, keine Rechenoperation durchzuführen, sondern **Text auszugeben**, der nicht berechnet wird.

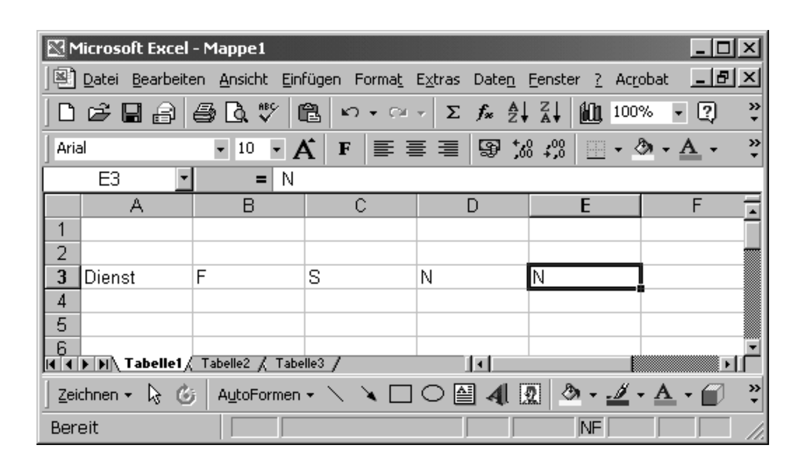
Mit diesem Beispiel wird deutlich, dass der Befehl WENN eine wichtige Funktion innerhalb Microsoft Excel besitzt.



07.2 Abhängiges Design

Microsoft Excel bietet die Möglichkeit in Abhängigkeit von Werten das Design einer oder mehrer Zellen zu verändern. Diese Funktion wird als „Bedingte Formatierung“ bezeichnet und findet sich im Menüpunkt „Format“. Beachten Sie, dass die bedingte Formatierung, wie viele Funktionen in Microsoft Excel sich auf die markierte/n Zelle/n beziehen.

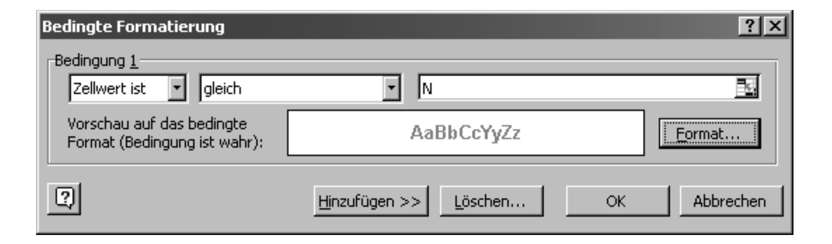
Um beispielsweise dafür zu sorgen, dass in einem Dienstplanmodell alle Nachtdienste (symbolisiert durch den Buchstaben N) rot und in Fettdruck erscheinen gehen Sie wie folgt vor:



Geben Sie die Tabelle aus obigem Beispiel ein.

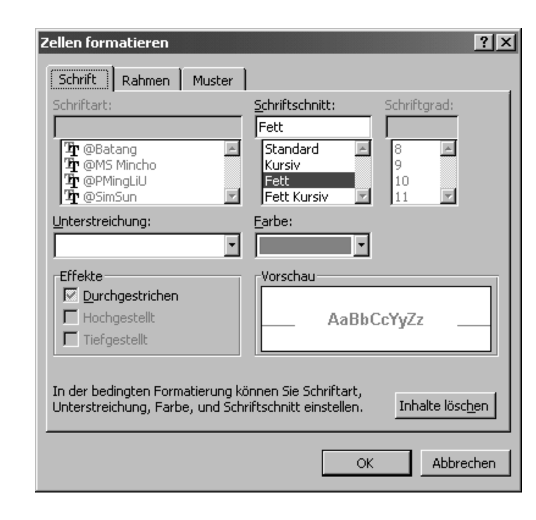
Markieren Sie anschließend den gewünschten Zellbereich in dem Sie auf den Spaltenkopf B bis E drücken, diese Spalten sind danach invertiert hinterlegt.

Im Menübefehl FORMAT -> BEDINGTE FORMATIERUNG sind nun folgende Eingaben zu machen:

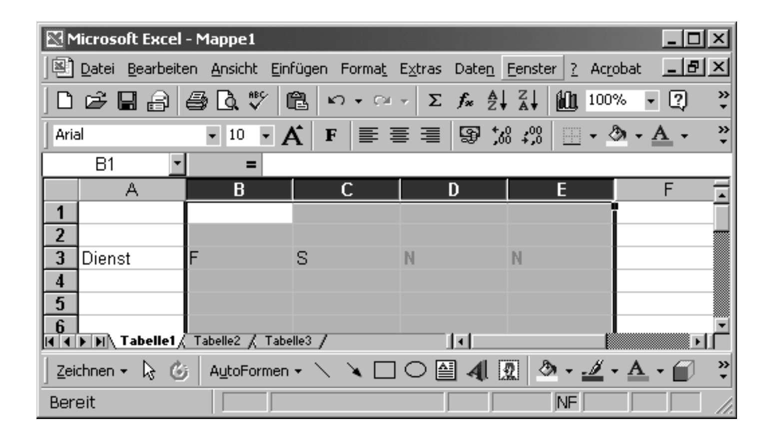


Bedingung 1 „Zellwert ist“ „gleich“ N

Klick auf FORMAT

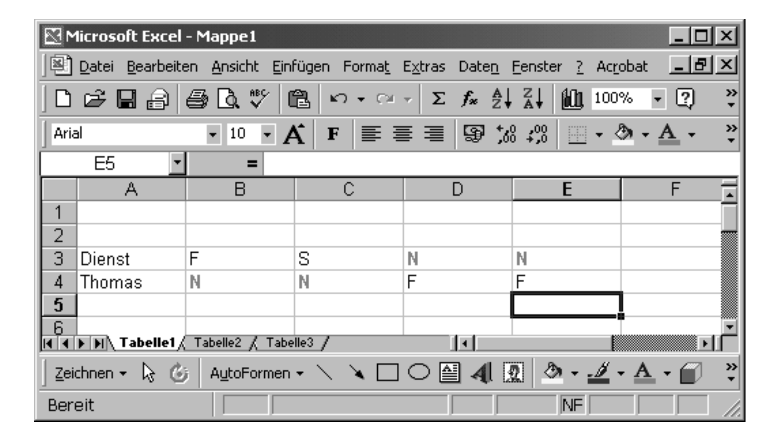


Schriftschnitt „Fett“ und Farbe „Rot“. Anschließend beide Fenster mit einem Klick auf „OK“ schließen.



Durch die Festlegung der bedingten Formatierung werden die Nachtdienst- Symbol rot geschrieben. Noch ist der Bereich B bis E invertiert hinterlegt, da er nach wie vor markiert ist. Durch einen Klick irgendwo in eine nicht markierte Zelle, wird die Markierung wieder aufgehoben.

Wird nun in der nächsten Zeile ein weiterer Mitarbeiter eingegeben, so gelten die Formatierungsbedingungen auch für diesen Eintrag:



Maximal lässt Microsoft Excel die Festlegung von **drei bedingte Formatierungen** für jede einzelne Zelle zu.

07.4 Verweise

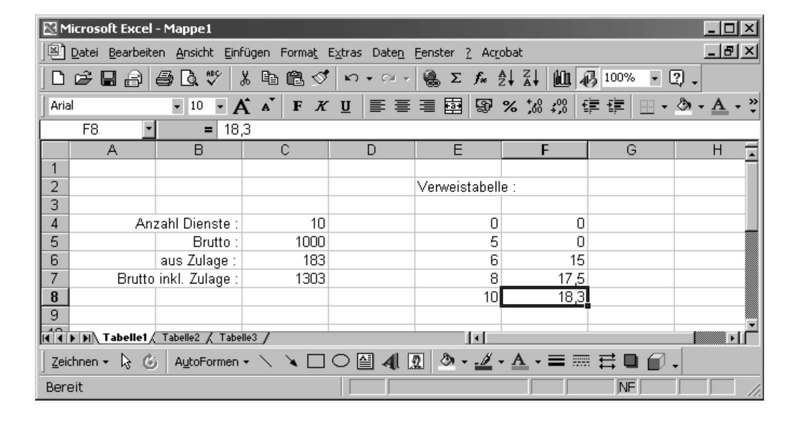
Es gibt in der Praxis Abhängigkeiten von Zahlenwerten, die sich nicht in einer linearen Berechnung beschreiben lassen. Angenommen in einer Dienstvereinbarung zu Nachtdiensten steht folgendes geschrieben:

Bei weniger als sechs Nachtdiensten ist keine Zulage zu zahlen, bei sechs oder sieben Diensten eine Zulage von 15 von Hundert des Bruttolohnes, bei acht oder neun Diensten 17,5 von Hundert des Bruttolohnes. Werden mehr als zehn Dienste im Monat absolviert, so ist eine Zahlung von 18,3 von Hundert und eine einmalige Zahlung von EUR 120 zu entrichten.

Es gibt zwischen den Zahlen kein erkennbares Verhältnis :

0,1,2,3,4,5 = keine Zulage
6,7 = 15 % vom Bruttolohn als Aufschlag
8,9 = 17,5 % vom Bruttolohn als Aufschlag
>= 10 = 18,3 % vom Bruttolohn + EUR 120

Um eine solche Berechnung zu realisieren, ist die Mischung von Logikbefehlen und eine neue Funktion, der VERWEIS notwendig.



Bereiten Sie zunächst die Tabelle für dieses Beispiel vor :

- Zelle B4 Eingabe „Anzahl Dienste :“ Ausrichtung linksbündig
- Zelle B5 Eingabe „Brutto :” Ausrichtung linksbündig
- Zelle B6 Eingabe „aus Zulage :” Ausrichtung linksbündig
- Zelle B7 Eingabe „Brutto inkl. Zulage” Ausrichtung linksbündig
- Zelle C4 enthält die Anzahl von Diensten ; hier 10
- Zelle C5 Bruttogehalt ; hier 1000
- Zelle E2 enthält einen Titel „Verweistabelle“
- Verweistabelle E4 Wert 0, F4 Wert 0
- Verweistabelle E5 Wert 5, F5 Wert 0
- Verweistabelle E6 Wert 6, F6 Wert 15
- Verweistabelle E7 Wert 8, F7 Wert 17,5
- Verweistabelle E8 Wert 10, F8 Wert 18,3

Im nächsten Schritt müssen wir Microsoft Excel dazu bringen aufgrund der Eingabe von Anzahl von Diensten in Zelle C4 in der Verweistabelle den entsprechenden Prozentwert zu suchen, der in die Berechnung einfließen soll.

Der Funktion in Zelle C6 lautet :

=VERWEIS (C4;E4:F8) * C5 / 100

Suchwert Verweistabelle [Suchkriterium | Ergebnisvektor]

Die angehängte Berechnung „* C5 /100“ sorgt für eine Prozentberechnung vom Bruttolohn ausgehend. Microsoft Excel sucht nun eine Zahl im Bereich E4 bis E8 die dem eingegebenem Wert in C4 entspricht und gibt die nebenstehende Zahl auf F4 bis F8 als Ergebnis zurück. Dieser Wert wird mit dem Bruttolohn aus C5 multipliziert und das komplette Ergebnis anschließend durch 100 dividiert.

Fehlt noch die Einmalzahlung von € 120, sofern mehr als zehn Dienste absolviert werden. Diese Anforderung wird mit dem Befehl WENN in der Ergebnisberechnung in C7 realisiert :

Fehlt noch die Einmalzahlung von € 120, sofern mehr als zehn Dienste absolviert werden. Diese Anforderung wird mit dem Befehl WENN in der Ergebnisberechnung in C7 realisiert :

=C5+C6+WENN(C4>=10;120;0)

Es wird in C7 der Bruttolohn aus C5 mit dem prozentuellen Ergebnis aus der Zulagenberechnung in C6 addiert. Zusätzlich wird der Wert 120 addiert, sofern C4 größer oder gleich zehn ist, beziehungsweise es wird 0 addiert.

Die Technik des Verweises ist beim ersten Kontakt sicherlich verwirrend, sodass ein zweites Beispiel beim Verständnis für dieses Gebiet helfen soll – da diese Möglichkeit einfach für viele Aufgaben in der Praxis unumgänglich ist.

Im zweiten Beispiel geht es um die Preisgestaltung beim Eintritt in eine Badeanstalt, die Preisstaffelung am Eingang liest sich wie folgt :

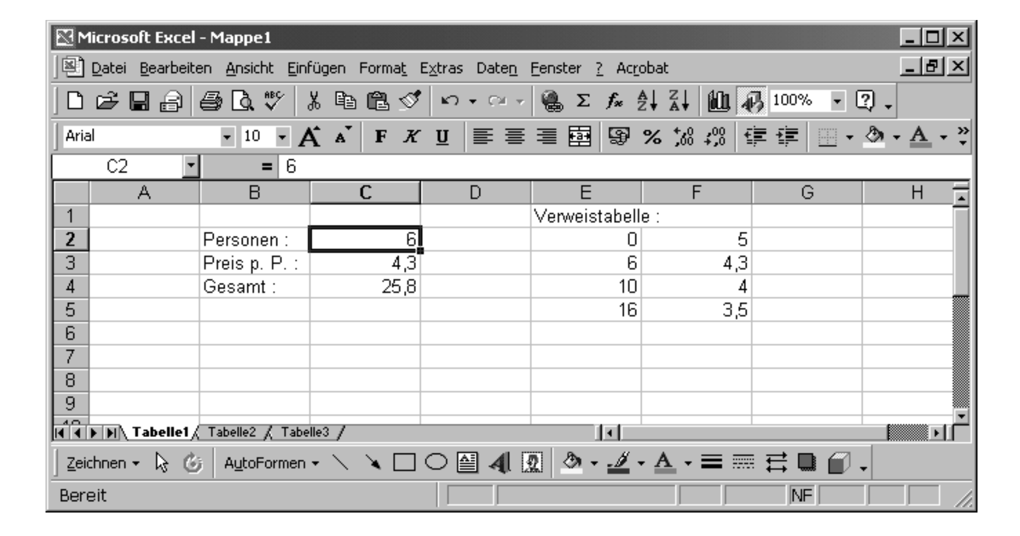
Einzelpersonen zahlen für den Eintritt € 5, Gruppen mit mehr als 6 Personen zahlen je Person € 4,30, Gruppen mit mehr als 9 Personen € 4,00 und Großgruppen mit mehr als 15 Personen zahlen pro „Nase“ € 3,50.

Um dem Kassierer das Leben einfach zu machen erstellen Sie eine Microsoft Excel Tabelle zur Berechnung des Eintrittspreises. Die Verweistabelle wird somit wie folgt aussehen :

0	5
6	4,30
10	4
16	3,50

Denn 0 bis 6 Personen zahlen jeweils € 5,00, 7 bis 8 Personen zahlen € 4,30, 10 bis 15 Personen € 4,00 und 16 und mehr Personen zahlen € 3,50.

Somit ergibt sich der spätere **Gesamtpreis** aus der Multiplikation der Anzahl von Personen mit dem aus der Verweistabelle ausgelesenen „Pro Personen Preis". Der Kassierer muss nun lediglich die Anzahl von Personen eingeben und erhält den zu berechnenden Preis.



Die Funktion in C3 lautet:

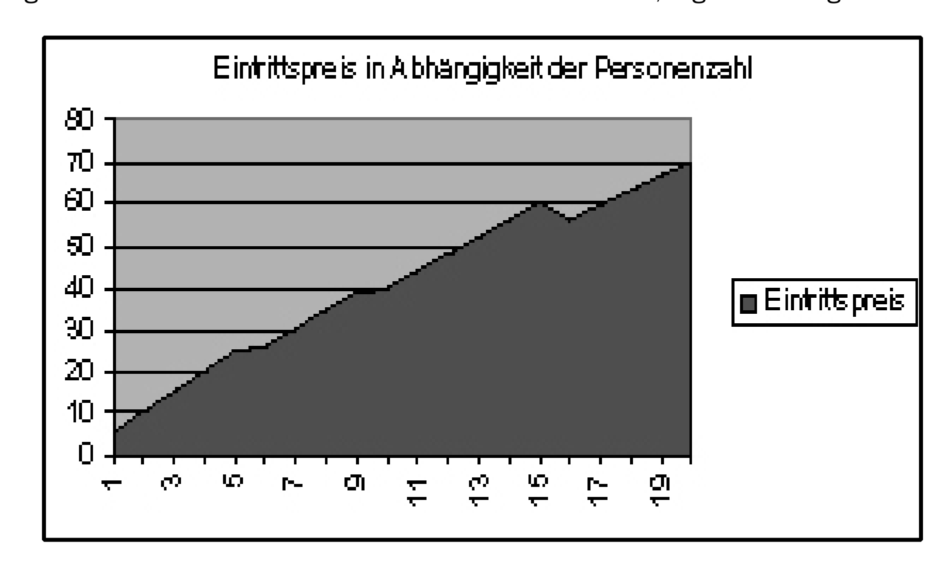
= VERWEIS (C2;E2:F5)

Die Funktion in C4 lautet:

= C2 * C3

Wie Sie später sehen können ist es gemäß dieser Preisstaffelung günstiger mit 16 Leuten in das Freibad zu gehen, als mit 15 Personen.

Wird der Diagramm- Assistent mit dem Zahlenmaterial konfrontiert, ergibt sich folgende Preistabelle:



08 Berechnungen mit Datum und Uhrzeit

Ein enormes Hilfsmittel ist Microsoft Excel bei der Berechnung von Datum und Uhrzeit. Mit Excel lassen sich **exakte Zeiträume** zwischen zwei Datumseinträgen berechnen, eine wichtige Grundlage zur Stundenberechnung und Dienstplangestaltung mit Microsoft Excel. Wie Datum- und Uhrzeitwerte eingegeben werden müssen, haben Sie bereits im Abschnitt 3 erfahren.

Aufgrund der Erfahrung, die der Autor als Dozent im Unterricht sammeln konnte, ist ein Umstand von Microsoft Excel zunächst schwer zu erfassen – und zwar der Beginn der **Zeitrechnung** im Programm.

Was ist denn eigentlich ein Datum in der westlichen Hemisphäre? Der Abstand in Tagen, Monaten und Jahren vom berechneten Tag vom Geburtstag von Jesus Christus. Genau genommen also ein gesetzter Tag X in der Geschichte der Menschheit. Andere Kulturen haben andere **Datumsberechnungsgrundlagen**, beispielsweise den Geburtstag des Propheten Mohammed. Nur in einer Tatsache sind sich alle Datumssysteme einig: ein Tag wird in „Pi mal Daumen" 24 Stunden zu je 60 Minuten zu je 60 Sekunden unterteilt. Dies ist eine Naturkonstante!

Die Programmierer von Microsoft Excel haben jedoch nicht den Ersten Januar des Jahres Null als **Tag EINS** gewählt, sondern den 1. Januar des Jahres 1900. Dies ist für die praktischen Berechnungen der heutigen Zeit kein Nachteil, es sei denn ein Mensch ist beispielsweise vor dem 1.1.1900 geboren – ein statistisch, aktuell gesehen, seltenster Fall.

Im Hintergrund berechnet Microsoft Excel alle Datumseingaben von diesem Tag ausgehend in eine reguläre Zahl um, dies soll ein Beispiel verdeutlichen:

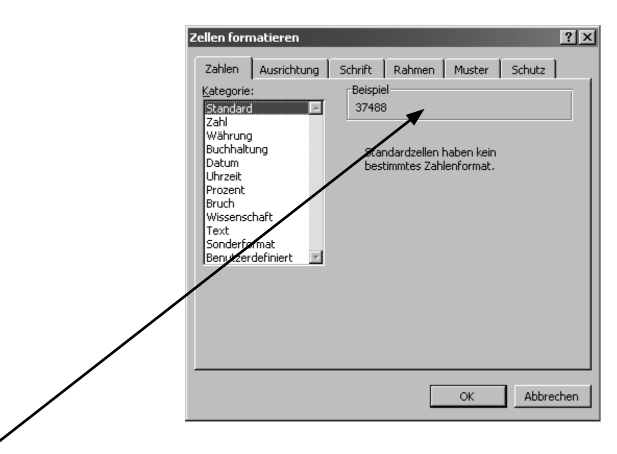
Geben Sie einfach ein Datum in eine Microsoft Excel Tabelle ein und zwar (siehe Abschnitt 3) in folgendem Format:

20.08.2002
oder anders ausgedrückt : TagTag Punkt MonatMonat Punkt JahrJahrJahrJahr

Wird die Zahl rechtsbündig in die Zelle eingetragen, so können Sie sichergehen, dass Microsoft Excel die Eingabe als Datum interpretiert hat. Nun soll in der Zelle C2 genau der gleiche Wert erscheinen:

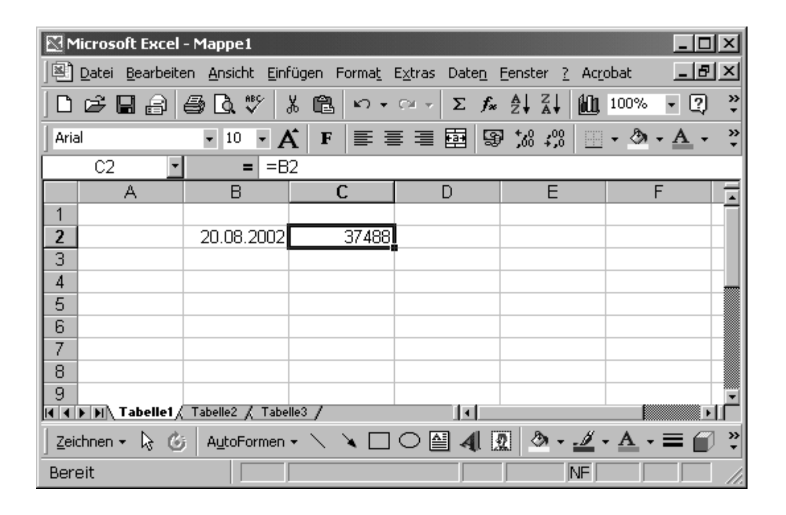
= B2

Wie Sie erkennen können, übernimmt Microsoft Excel das Format automatisch auf die Nachbarzelle. Nun soll Microsoft Excel diese Zahl einmal so darstellen, wie sie im Hintergrund für Microsoft Excel tatsächlich aussieht. Stellen Sie dazu das Format in der Zelle C2 auf eine Standardzahl um. Dazu öffnen Sie das Menü „Format", klicken auf „Zellen" und wählen „Standard" :

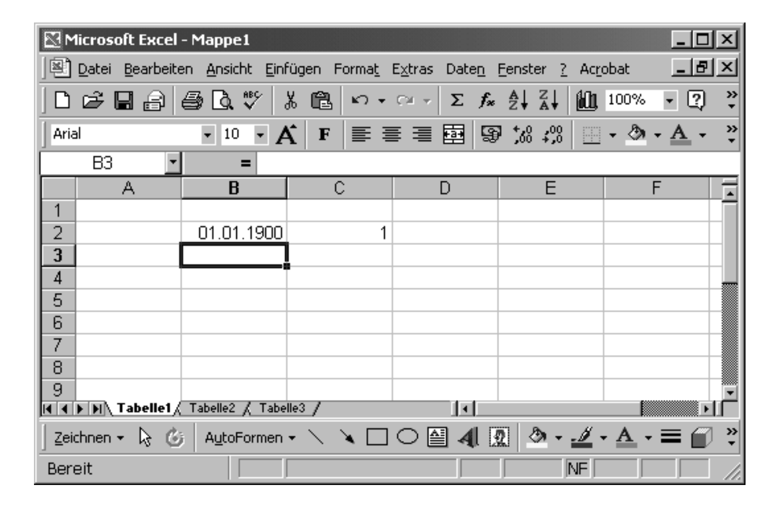


Wie Sie im Kasten Beispiel bereits sehen können, ist die serielle Zahl des Datums 20.08.2002 genau 37488.

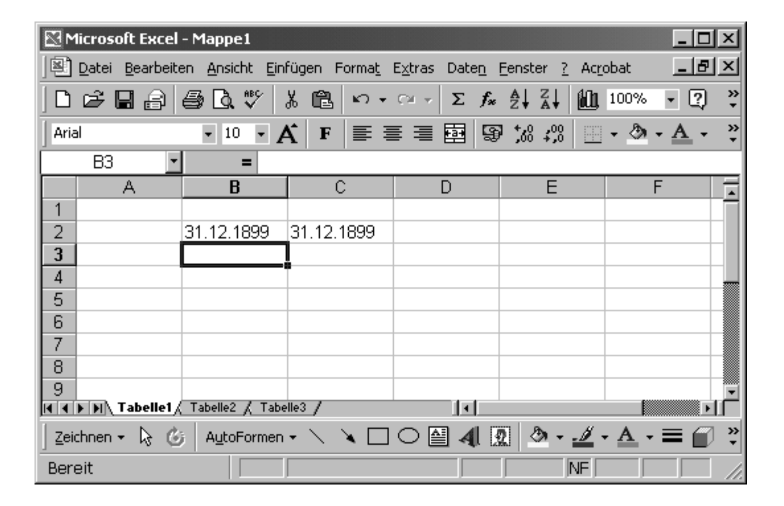
Das bedeutet, das Datum 20.08.2002 ist vom Tag Eins der Zeitberechnung in Microsoft Excel 37488 Tage entfernt.



Geben Sie anstelle des 20.08.2002 in die Zelle B2 das Datum 01.01.1900 ein, so wird dies auch bei der Umsetzung der Zelle C2 deutlich:



Versuchen Sie nun den 31.12.1899 und es wird in der Zelle C2 keine serielle Zahl mehr berechnet, sondern der Texteintrag aus B2 übernommen. Zur Erinnerung – es können keine Datumsberechnungen vor dem 01.01.1900 vorgenommen werden!



08.1 Rechnen mit Uhrzeiten

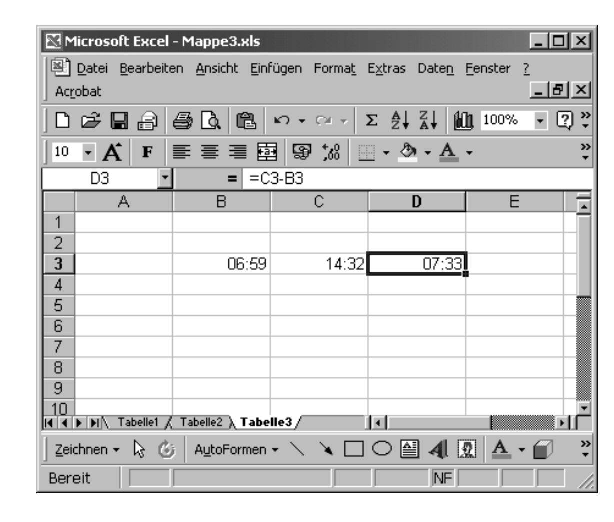
Ebenso einfach wie die Berechnung von Datumseingaben ist die Berechnung von Uhrzeiten unter Microsoft Excel. Möchten Sie beispielsweise die Dauer eines Arbeitsdienstes errechen so ist in erster Linie wieder die Art der Eingabe des Wertes wichtig.

Zeitangaben erwartet Microsoft Excel im Format

StundeStunde : MinuteMinute : SekundeSekunde

Für ein Beispiel geben Sie in eine leere Tabelle in die Zelle B3 06:59 ein und in die Zelle C3 den Wert 14:32. Wenn nun errechnet werden soll, wie viel Stunden und Minuten seit dem Eintrag vergangen sind lautet eine Funktion in D3:

= B3 – C3



Das Ergebnis ist 07:33 Stunden/Minuten. In der Standardeinstellung ignoriert Microsoft Excel die Sekunden und setzt sie automatisch auf 00, sofern vom Benutzer kein zweiter Doppelpunkt mit Sekundenangabe angegeben wurde.

Verbinden wir nun dieses Beispiel mit einer Bruttolohnberechnung. Die Person hat einen Bruttolohn von € 19 je Stunde:

Somit würde D4 lauten können:

= D3*19

Nun ist das produzierte Ergebnis von 23:17 alles andere als eine befriedigende Entlohnung. Microsoft Excel hat zum einen das Format „Uhrzeit“ mit in das Ergebnis genommen (lässt sich Umstellen durch Markierung der Zelle D4 mit einem Rechtsklick, im Kontextmenü „Zelle formatieren“ Format auf Zahl einstellen). Aber auch dann ist die Entlohnung mit 5,98 für den Mitarbeiter wenig lohnenswert.

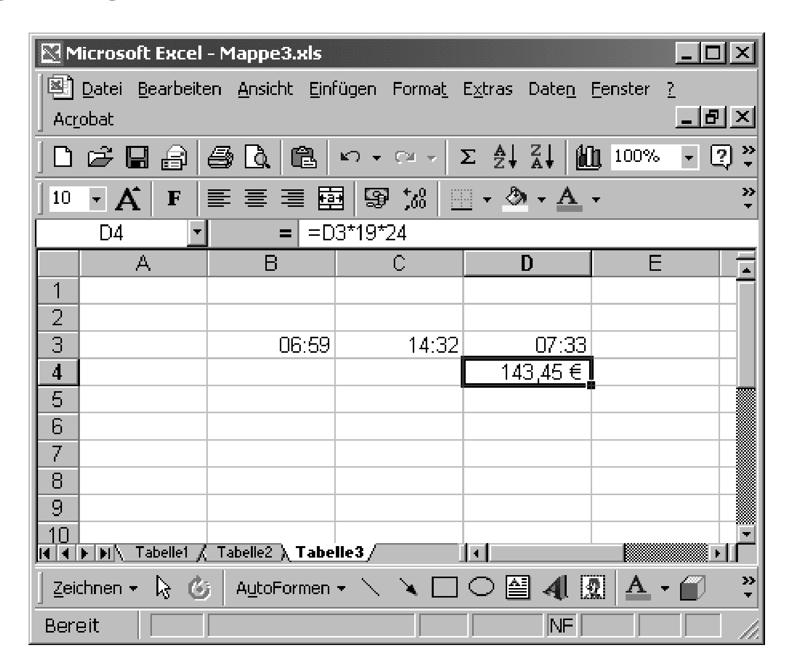
Wie kommt es zu diesem Verhalten von Microsoft Excel?

Der Ursprung des Verhalten liegt in der Art und Weise begründet, wie Microsoft Excel den Datums- & Uhrzeitberechnungen angeht. Wie bereits im vorherigen Abschnitt erörtert beginnt die Zeitrechnung für Microsoft Excel am 01.01.1900. Somit ist eine Stunde für Microsoft Excel der 24te Teil des Tages. Das Ergebnis ist somit das Produkt der Berechnung Teil des Tages multipliziert mit dem Stundensatz € 19.

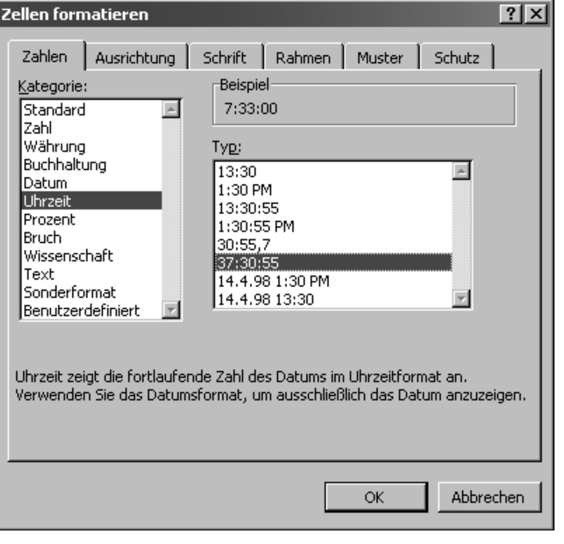
Um hier eine korrekte Bruttolohnerrechnung zu erzielen ist die anschließende Multiplikation mit 24 unumgänglich, somit lautet die Funktion in D4:

=D3 * 19 * 24

Das Bruttolohnergebnis liegt somit bei € 143,45.



Um eine Stundezahl, die größer als 24 ist korrekt anzeigen zu können muss im Menü FORMAT -> ZELLEN -> UHRZEIT auf 37:30:55 eingestellt werden. Oder das benutzerdefinierte Format [h]:mm , um die Sekundenangabe auszublenden.



08.2 Aufenthaltsdauerberechnung

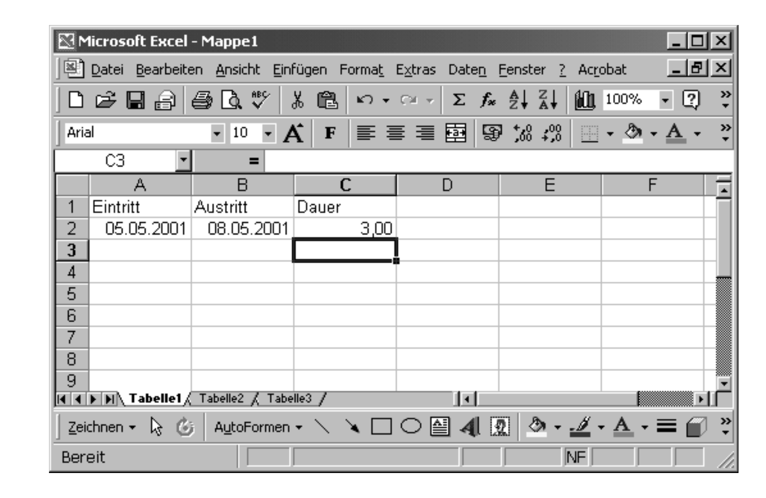
Die gewonnenen Erkenntnisse sollen nun in ein praktisches Beispiel münden, einem Aufenthaltsdauerberechner. Bereiten Sie dazu folgende Tabelle vor:

Um den Abstand zwischen beiden Datumsangaben zu berechnen wird einfach der Wert von A2 von B2 abgezogen. Somit lautet die Funktion in C2:

= B2 – A2

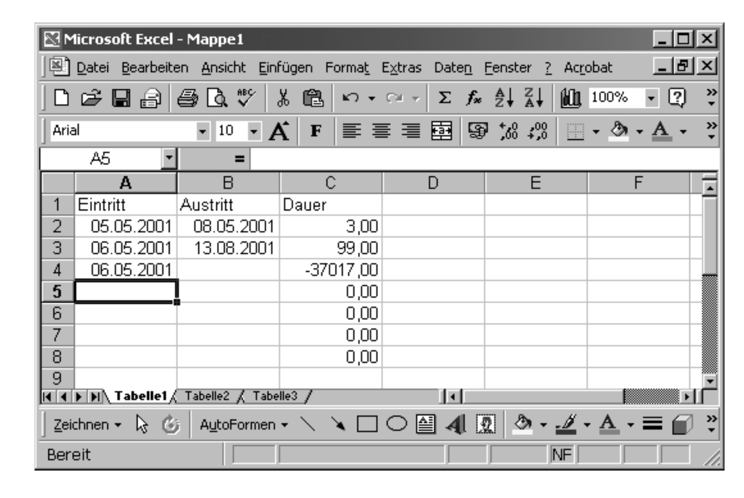
Nun erscheint als Ergebnis jedoch etwas zunächst Seltsames: 03.01.1900 ! Das erklärt sich im System der Formatübernahme von Microsoft Excel. Wird ein Datum von einem Datum abgezogen, geht Microsoft Excel davon aus, dass auch das Ergebnis im gleichen Format angeliefert werden soll: als Datum. Erst wenn Sie die Zelle C2 über FORMAT -> ZELLE -> STANDARD in eine Standardzahl verwandeln, erscheint das gewünschte Ergebnis 3.

Aber auch der Eintrag 03.01.1900 hat einiges genau ausgesagt, nämlich drei Tage, keinen Monat und kein Jahr vom Tag Eins der Microsoft Excel Datumsberechnung aus gesehen.



Wird nun die Tabelle erweitert, setzen ältere Versionen von Microsoft Excel leider nicht automatisch die Funktion in die Spalte C ein. Dies muss der Anwender von Hand tun, in dem das Ausfüllkästchen nach unten „gezogen“ wird.

Für einen praktischen Einsatz wäre es von Vorteil, wenn Kollegen einfach nur das Datum eintragen müssen, und es wird automatisch weitergerechnet. Somit ist es am Tabellen- Designer, die Tabelle dahingehend vorzubereiten. Dies kann dadurch geschehen, dass mit dem Ausfüllkästchen die Tabelle auf rund 100 Einträge, oder mehr vorbereitet wird.



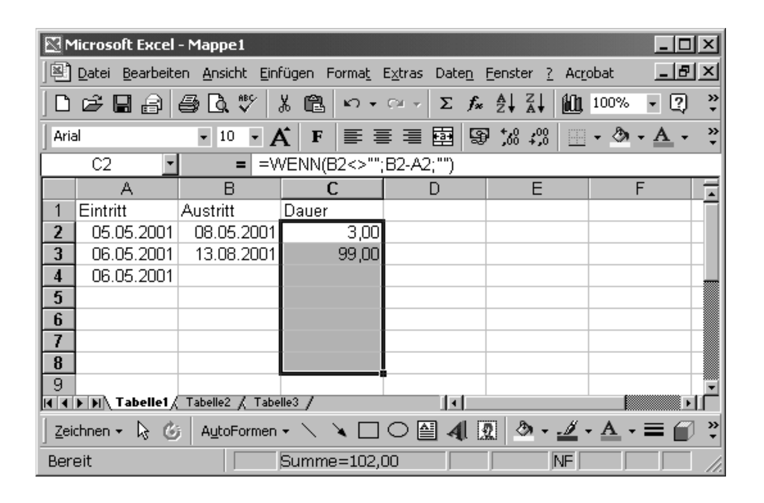
	A	B	C
1	Eintritt	Austritt	Dauer
2	05.05.2001	08.05.2001	3,00
3	06.05.2001	13.08.2001	99,00
4	06.05.2001		-37017,00
5			0,00
6			0,00
7			0,00
8			0,00

Dies hat jedoch zur Folge, dass beispielsweise Patienten, die bereits ein Eintrittsdatum haben, aber noch kein Austrittsdatum, mit Minuswerten berechnet werden. Dies ist in der Praxis unbrauchbar!

Sie haben im vorherigen Abschnitt einen Befehl kennen gelernt, mit dem hier Abhilfe geschaffen werden kann -> dem WENN Befehl. Wir müssen der Tabelle beibringen, nur dann die Berechnung durchzuführen, wenn auch ein Austrittsdatum eingetragen wurde.

So lautet die Funktion für Zelle C2 :

= WENN(B2<>"";B2-A2;"")



=WENN(B2<>"";B2-A2;"")

	A	B	C
1	Eintritt	Austritt	Dauer
2	05.05.2001	08.05.2001	3,00
3	06.05.2001	13.08.2001	99,00
4	06.05.2001		

Die obige Funktion lautet im Klartext:

Ist der Wert der Zelle B2 kleiner oder größer NICHTS, dann ziehe vom Wert B2 den Wert A2 ab, ist das nicht der Fall schreibe in die Zelle C2 NICHTS.

Die Betonung NICHTS kommt nicht von ungefähr, denn in der Programmierung wird zwischen NICHTS und Null unterschieden. Die Zahl Null ist ein existierender Wert auf dem Zahlenstrahl, Nichts hingegen ist einfach überhaupt nicht vorhanden.

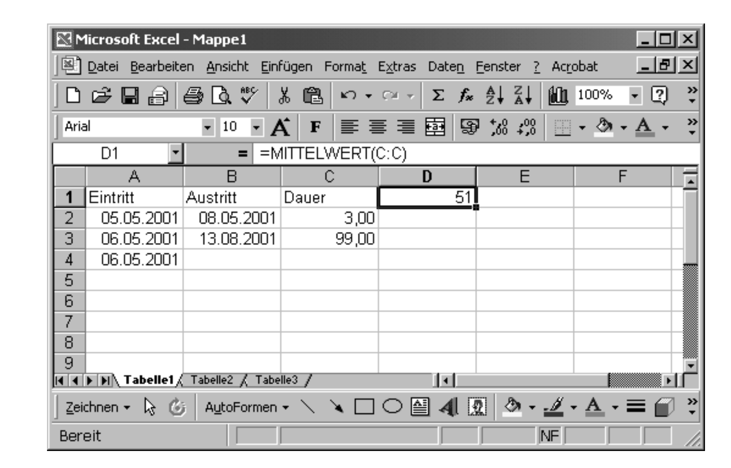
Praktisch hat dies für die folgende Weiterbearbeitung folgenden Hintergrund: Die Tabelle soll nun die durchschnittliche Liegedauer von Patienten aus dem Jahrbuch berechnen. Da wir aber eine unbekannt große Anzahl von Patientendatensätzen haben (wobei sich dies mit Microsoft Excel sehr wohl ermitteln ließe) verwenden wir im folgenden Beispiel eine neue Funktion, und zwar MITTELWERT, um der Durchschnitt zu berechnen.

08.3 Durchschnittsberechnung

Was genau ist denn mathematisch beschrieben der Durchschnitt? Die Summe aller Werte geteilt durch die Anzahl von einzelnen Werten. Diese Funktion ist in Microsoft Excel bereits enthalten, im Befehl MITTELWERT.

Für die Beispieltabelle aus Abschnitt 8.2 wird nun im Feld D1 der Mittelwert aller Aufenthalte berechnet:

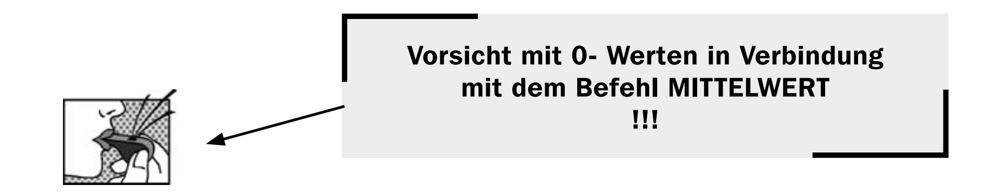
= MITTELWERT(C:C)



=MITTELWERT(C:C)

	A	B	C	D
1	Eintritt	Austritt	Dauer	51
2	05.05.2001	08.05.2001	3,00	
3	06.05.2001	13.08.2001	99,00	
4	06.05.2001			

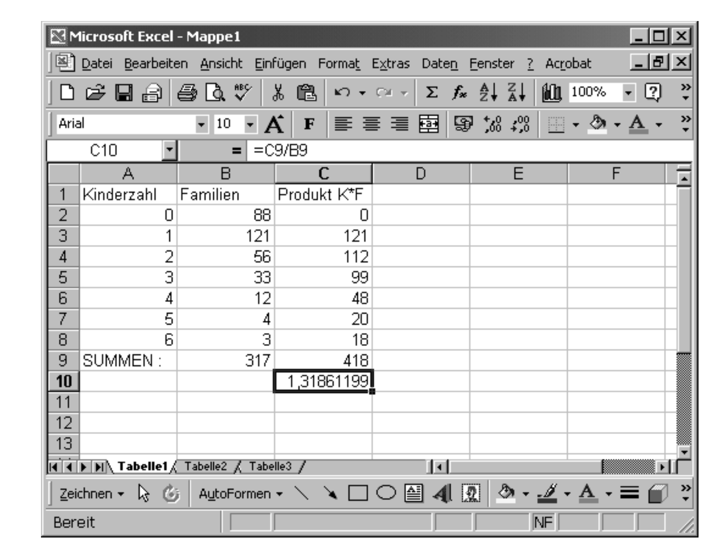
Wären die Nullwerte nicht durch den WENN- Befehl aus Abschnitt 8.2 entfernt worden, so wäre ein komplett anderes Ergebnis herausgekommen. Da die Zahl 0 die Anzahl der Werte nach oben bewegt, obwohl sie auf die Gesamtsumme keinen Einfluss nehmen.



Über den Sinn oder Unsinn einer Durchschnitts- Berechnung lässt sich grundsätzlich streiten, da es aus dem Bereich der Statistik kommt soll ein weitere Gedanken hier kurz erwähnt werden.

„Stellen Sie sich vor sie haben auf dem Spielplatz 10 Kinder sitzen, von denen 9 gar kein Geld dabei haben und einer einen 100 Euro Schein. Somit hat, gemäß der einfachen Durchschnittsberechung, jedes Kind 10 Euro. Das dieses Modell niemanden gerecht wird ist klar, den der Eine, der 100 Euro bei sich trägt, hat statistisch 90 Euro zuviel und die neun Kinder die gar kein Geld haben, haben jeweils 10 Euro zuviel."

Noch ein Beispiel zur Berechnung von Durchschnittswerten. Es soll die durchschnittliche Kinderzahl berechnet werden:


C10 =C9/B9

	A	B	C
1	Kinderzahl	Familien	Produkt K*F
2	0	88	0
3	1	121	121
4	2	56	112
5	3	33	99
6	4	12	48
7	5	4	20
8	6	3	18
9	SUMMEN :	317	418
10			1,31861199

Im obigen Beispiel wird im Bereich C2 bis C8 das Produkt (B2*A2, B3*A3...) gebildet und in B9 die Summe aus allen in der Umfrage teilnehmenden Familien (=SUMME(B2:B8)) und aus der Berechnung der Produkte in C9 (=SUMME(C2:C8)) ermittelt.

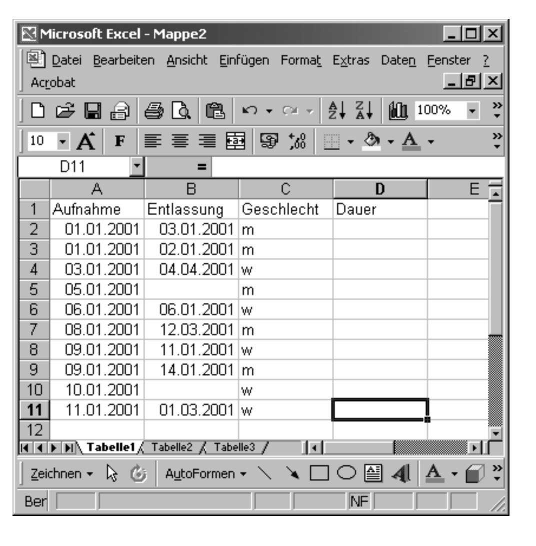
Um nun die durchschnittliche Kinderzahl zu ermitteln, wird in C10 das Ergebnis aus dem Produkt von C9 durch die Anzahl der Familien in B9 geteilt. In diesem Beispiel ist die Funktion MITTELWERT von Microsoft Excel unbrauchbar, da der Mittelwert auf C2 bis C8 59,71 ergeben würde. Das wäre die mittlere Verteilung aller Produkte aus Kinderanzahl und Familie.

08.4 Zählen von Merkmalen

Viele Dinge im täglichen Leben erfordern die Arbeit mit der Tabellenkalkulation aber nicht das direkte Rechnen mit Zahlenwerten, sondern mit Buchstaben als Symbol für einen Wert (beispielsweise ein Dienstplan).

In einem ersten Beispiel soll ein Jahrbuch, in dem neben dem Aufnahmedatum, Entlassungsdatum auch das Geschlecht hinterlegt werden. Beachten Sie für dieses Beispiel den Abschnitt 8.3 – Durchschnittsberechnung.

Zunächst sollen folgende Informationen in eine Tabelle eingeben werden, alle nötigen Funktionen werden zum späteren Zeitpunkt eingebaut:

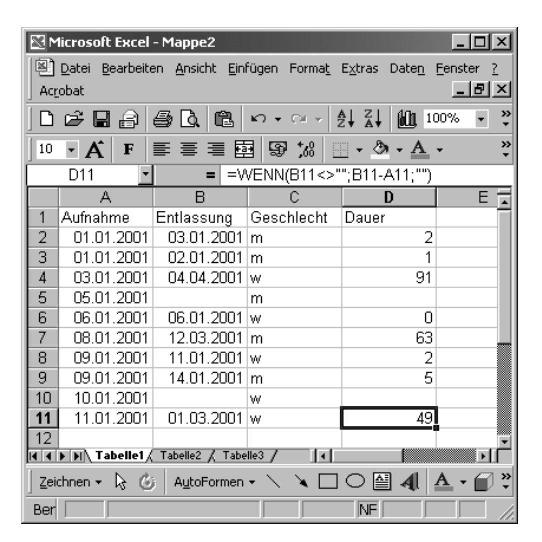


	A	B	C	D
1	Aufnahme	Entlassung	Geschlecht	Dauer
2	01.01.2001	03.01.2001	m	
3	01.01.2001	02.01.2001	m	
4	03.01.2001	04.04.2001	w	
5	05.01.2001		m	
6	06.01.2001	06.01.2001	w	
7	08.01.2001	12.03.2001	m	
8	09.01.2001	11.01.2001	w	
9	09.01.2001	14.01.2001	m	
10	10.01.2001		w	
11	11.01.2001	01.03.2001	w	

Um nun die Aufenthaltsdauer zu errechnen, wird eine Funktion benötigt, die bereits im vorherigen Abschnitt erläutert wurde:

Für D2 lautet sie: = WENN(B2<>"";B2-A2;"")

Das Format der Zelle ist über Format -> Zellen auf Standard zu stellen.


D11 =WENN(B11<>"";B11-A11;"")

	A	B	C	D
1	Aufnahme	Entlassung	Geschlecht	Dauer
2	01.01.2001	03.01.2001	m	2
3	01.01.2001	02.01.2001	m	1
4	03.01.2001	04.04.2001	w	91
5	05.01.2001		m	
6	06.01.2001	06.01.2001	w	0
7	08.01.2001	12.03.2001	m	63
8	09.01.2001	11.01.2001	w	2
9	09.01.2001	14.01.2001	m	5
10	10.01.2001		w	
11	11.01.2001	01.03.2001	w	49

Für das Ergebnis in D6 ergibt sich nun bereits ein Problem, ein Patient, der am gleichen Tag geht, wie er kommt ist nach dieser Statistik 0 Tage da und verändert das Durchschnittsergebnis der Einrichtung. Ob ein solcher Wert als 1 gerechnet werden soll, ist Definitionssache der Einrichtung. Für diesen Fall lautet die Funktion für D2:

=WENN(B2<>"";B2-A2+WENN(B2=A2;1;0);"")

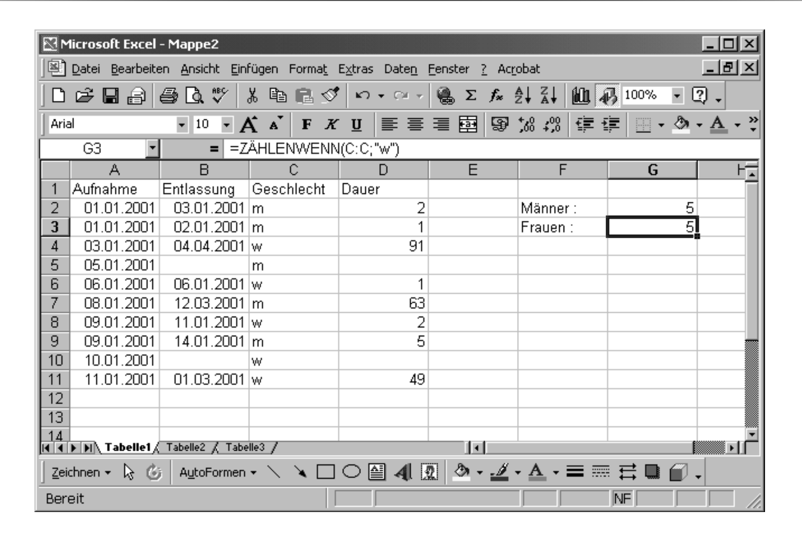
Die Verschachtelung des zweiten WENN- Befehls sorgt für die Addition mit dem Wert 1, wenn Entlass- und Aufnahmedatum gleich sind.

D11 =WENN(B11<>"";B11-A11+WENN(B11=A11;1;0);"")

	A	B	C	D
1	Aufnahme	Entlassung	Geschlecht	Dauer
2	01.01.2001	03.01.2001	m	2
3	01.01.2001	02.01.2001	m	1
4	03.01.2001	04.04.2001	w	91
5	05.01.2001		m	
6	06.01.2001	06.01.2001	w	1
7	08.01.2001	12.03.2001	m	63
8	09.01.2001	11.01.2001	w	2
9	09.01.2001	14.01.2001	m	5
10	10.01.2001		w	
11	11.01.2001	01.03.2001	w	49

Nun soll die Verteilung der Geschlechter in dieser Tabelle prozentuell dargestellt werden, dazu muss Microsoft Excel die Gesamtsumme aller Einträge in das Verhältnis zu Geschlechtsmerkmal „m“ und „w“ stellen.

Ein neuer Befehl von Microsoft Excel ist für diesen Vorgang besonders hilfreich : ZÄHLENWENN !



	A	B	C	D	E	F	G
1	Aufnahme	Entlassung	Geschlecht	Dauer			
2	01.01.2001	03.01.2001	m	2		Männer :	5
3	01.01.2001	02.01.2001	m	1		Frauen :	5
4	03.01.2001	04.04.2001	w	91			
5	05.01.2001		m				
6	06.01.2001	06.01.2001	w	1			
7	08.01.2001	12.03.2001	m	63			
8	09.01.2001	11.01.2001	w	2			
9	09.01.2001	14.01.2001	m	5			
10	10.01.2001		w				
11	11.01.2001	01.03.2001	w	49			

Für die Zelle G2 („Männer") lautet die Funktion:

= ZÄHLENWENN (C:C;"m") „Zähle alle Zellen der Spalte C die klein M lauten"

Bereich Suchkriterium

Die Anführungszeichen um das kleine M besagen Microsoft Excel nur nach einem Text zu suchen, der genau so lautet. Analog lautet die Formel in G3 („Frauen"):

= ZÄHLENWENN(C:C;"w")

Das nun vorliegende Ergebnis ist jedoch kein Prozentwert, sondern der absolute Wert, der nun in ein prozentuelles Verhältnis verwandelt werden soll. Da zunächst die Gesamtanzahl von Männern und Frauen in der Tabelle errechnet werden muss, wird die Funktion komplexer (G2):

= ZÄHLENWENN(C:C;"m") * 100 / (ZÄHLENWENN(C:C;"m")+ZÄHLENWENN(C:C;"w"))

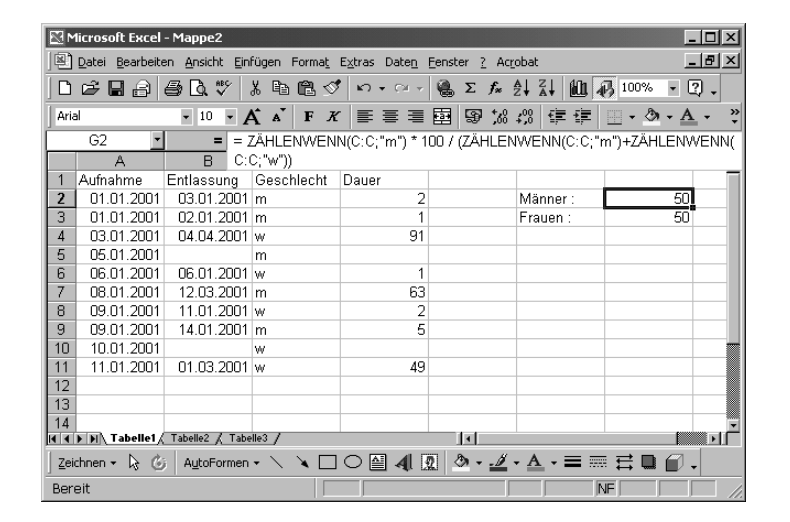
Mathematisch steckt in dieser Funktion eine einfache Drei- Satz Berechnung:

Gesamtsumme — 100 Prozent
Anzahl Merkmal — x

x = Anzahl Merkmal * 100 : Gesamtsumme

Analog lautet die Funktion für G3 :

= ZÄHLENWENN(C:C;"w") * 100 / (ZÄHLENWENN(C:C;"m")+ZÄHLENWENN(C:C;"w"))



	A	B	C	D	E	F	G
1	Aufnahme	Entlassung	Geschlecht	Dauer			
2	01.01.2001	03.01.2001	m	2		Männer :	50
3	01.01.2001	02.01.2001	m	1		Frauen :	50
4	03.01.2001	04.04.2001	w	91			
5	05.01.2001		m				
6	06.01.2001	06.01.2001	w	1			
7	08.01.2001	12.03.2001	m	63			
8	09.01.2001	11.01.2001	w	2			
9	09.01.2001	14.01.2001	m	5			
10	10.01.2001		w				
11	11.01.2001	01.03.2001	w	49			

Solange das Verhältnis 50 : 50 steht, sind die Prozentzahlen ordentlich dargestellt, sobald sich dieses Verhältnis ändert, produziert Microsoft Excel eine lange Kommazahl. Begrenzen Sie die Anzahl der Kommastellen mit Hilfe der Befehlskombination:
FORMAT -> ZELLEN -> ZAHL

Im nun folgendem Schritt soll die Aufenthaltsdauer von Männern und Frauen berechnet werden, da diese in diesem Beispiel sicherlich unterschiedlich ausfallen dürfte. Somit darf Microsoft Excel nur dann die Dauer durch die Anzahl der Nennung teilen, wenn das jeweilige Merkmal (weiblich oder männlich) erfüllt wurde.

Um dies zu bewerkstelligen ist zum einen ein neuer Befehl zu nennen, zum anderen muss in der gesamten Tabelle eine inhaltliche Veränderung vorgenommen werden. Diese Veränderung beruht auf folgendem Gedanken: Wenn dieser neue Befehl alle

Aufenthaltsdauern gemäß des Geschlecht in Spalte C berechnet, so rechnet er auch diese Einträge mit, die noch nicht entlassen wurden, und somit einen Wert von NICHTS (in dem Falle beziehungsweise Null) in der Aufenthaltsdauer haben. Das dies Auswirkungen auf den Durchschnitt hat, dürfte klar sein.

Es soll nun in die Tabelle eine **Spalte eingefügt** werden, die das Geschlechtsmerkmal nur dann trägt, wenn die Entlassung bereits vorgenommen wurde. Die Verteilung der Geschlechter in der Tabelle selbst soll von diesem Procedere jedoch unberührt bleiben.

Um eine **Spalte einzufügen** klicken Sie mit der rechten Maustaste auf den Spaltenkopf D und wählen Sie im Kontextmenü „Zellen einfügen" aus. Eine neue Spalte, mit dem Titel D wird eingefügt, alle rechtsliegenden Spalten werden um einen verschoben und die 255zigste Spalte am Ende der Tabelle gelöscht. Da sie keine Werte enthielt, hat Sie Microsoft Excel auch nicht mit einer Fehlermeldung vor diesem Vorgang gewarnt. Alle Funktionen wurden von Microsoft Excel automatisch so umgeschrieben, dass alle Berechnungen wieder korrekt durchgeführt werden.

Der Befehl WENN wird bei der „Ausmusterung der Geschlechtsmerkmale ohne Entlassungsdatum" (wie das klingt!) eingesetzt:

Für die Zelle D2 lautet die Funktion:

= WENN(B2<>"";C2;"")

Im Klartext bedeutet diese Funktion:

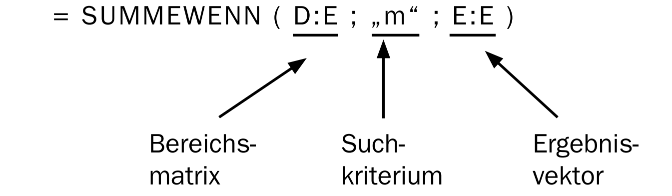
„Wenn in B2 ein Wert ist, schreibe in D2 den Wert aus D2, ansonsten schreibe in D2 Nichts".

Nun soll der **allgemeine Mittelwert** für beide Geschlechter in H6 errechnet in werden:

= MITTELWERT(E:E)

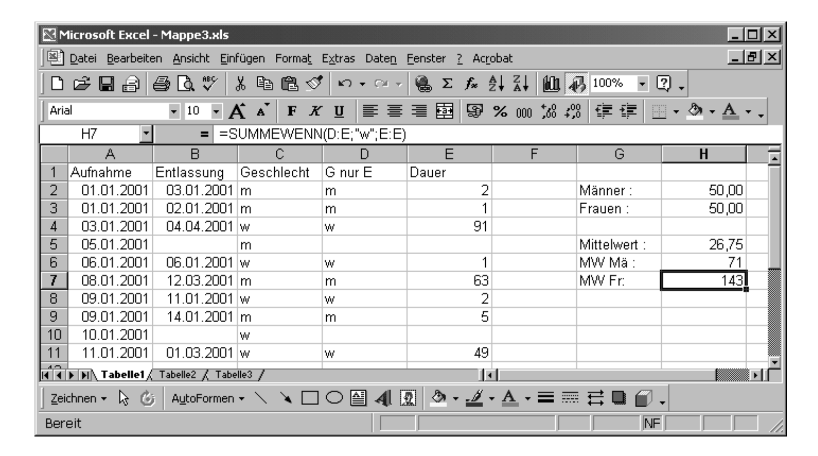
Um nun ausschließlich die Aufenthaltsdauer derer zu berechnen, der Geschlechtsmerkmal „m" ist, wird folgende Funktion benötigt:

= SUMMEWENN (D:E ; „m" ; E:E)



Analog dazu lautet die Funktion zur Berechnung der weiblichen Patienten in der Zelle H7 :

=SUMMEWENN(D:E;“w“;E:E)



H7 =SUMMEWENN(D:E;"w";E:E)

	A	B	C	D	E	F	G	H
1	Aufnahme	Entlassung	Geschlecht	G nur E	Dauer			
2	01.01.2001	03.01.2001	m	m	2		Männer :	50,00
3	01.01.2001	02.01.2001	m	m	1		Frauen :	50,00
4	03.01.2001	04.04.2001	w	w	91			
5	05.01.2001		m				Mittelwert :	26,75
6	06.01.2001	06.01.2001	w	w	1		MW Mä :	71
7	08.01.2001	12.03.2001	m	m	63		MW Fr.	143
8	09.01.2001	11.01.2001	w	w	2			
9	09.01.2001	14.01.2001	m	m	5			
10	10.01.2001		w					
11	11.01.2001	01.03.2001	w	w	49			

Wenn es nachgerechnet wird, so haben wir zumindest die Gewissheit, das Microsoft Excel die Werte korrekt berechnet hat. In einer Tabelle mit mehreren hundert Einträgen ist dieser Befehl ein wahrer Segen.

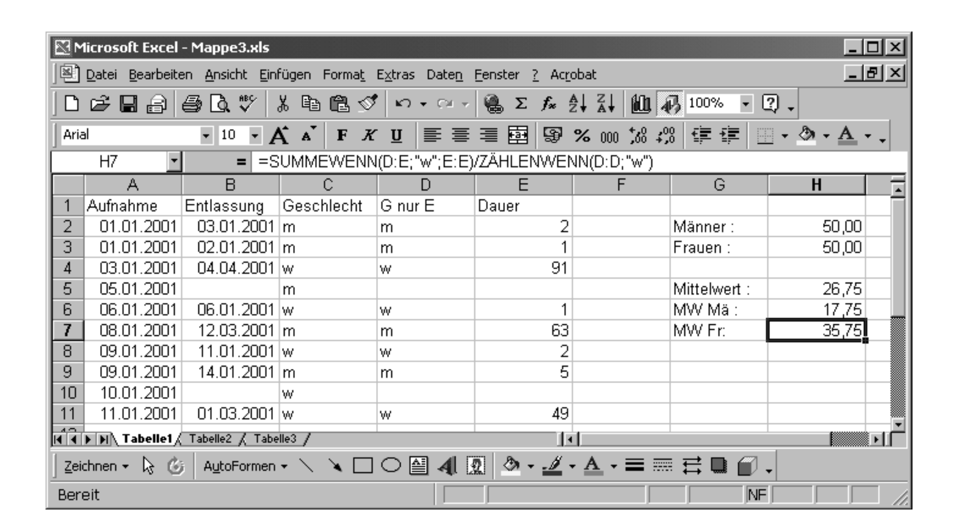
Nun wurde mit dem Befehl jedoch nicht die durchschnittliche Aufenthaltsdauer nach Geschlechtern errechnet, sondern die Summe der Gesamtdauer. Dieses Ergebnis ist noch durch die Anzahl der in die Berechnung eingegangenen Nennungen zu teilen. Zu diesem Zweck wurde die Spalte D neu eingefügt. Dies erledigt der Befehl ZÄHLENWENN, der bereits mehrfach eingesetzt wurde.

Somit verändert sich die Funktion in H6 zu:

=SUMMEWENN(D:E;“m“;E:E)/ZÄHLENWENN(D:D;“m“)

Und die Berechnung in H7 zu :

=SUMMEWENN(D:E;“w“;E:E)/ZÄHLENWENN(D:D;“w“)



H7 =SUMMEWENN(D:E;"w";E:E)/ZÄHLENWENN(D:D;"w")

	A	B	C	D	E	F	G	H
1	Aufnahme	Entlassung	Geschlecht	G nur E	Dauer			
2	01.01.2001	03.01.2001	m	m	2		Männer :	50,00
3	01.01.2001	02.01.2001	m	m	1		Frauen :	50,00
4	03.01.2001	04.04.2001	w	w	91			
5	05.01.2001		m				Mittelwert :	26,75
6	06.01.2001	06.01.2001	w	w	1		MW Mä :	17,75
7	08.01.2001	12.03.2001	m	m	63		MW Fr.	35,75
8	09.01.2001	11.01.2001	w	w	2			
9	09.01.2001	14.01.2001	m	m	5			
10	10.01.2001		w					
11	11.01.2001	01.03.2001	w	w	49			

Für eine fertige Tabelle, die in den praktischen Einsatz zu bringen ist, fehlen jedoch einige Sicherheitsvorkehrungen, die im nächsten Abschnitt erläutert werden. Auch müssen die Funktionen mit dem Ausfüllkästchen bis zu einem gewissen Punkt (evtl. 500) „heruntergezogen“ werden.

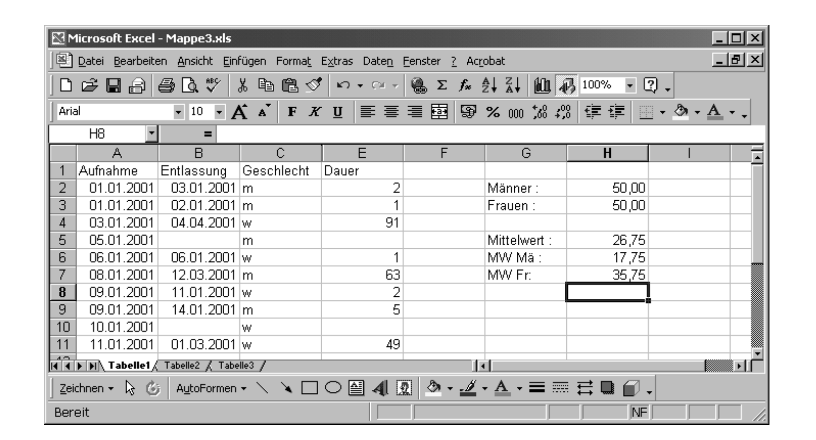
09 Tabellen vor Fehlern schützen

In diesem Abschnitt werden Methoden beschrieben, die vor Fehlern durch unachtsame Benutzer schützen. Besonders Funktionen sind vor dem unberechtigten Zugriff zu schützen, da Änderungen, die hier vorgenommen werden, sehr schwer zu finden sind.

09.1 Spalten oder Zeilen ausblenden

Eines der einfachsten Mittel, einen Bereich einer Tabelle vor dem Zugriff zu schützen, ist das ausblenden von Bereichen. Durch das Ausblenden wird der Bereich lediglich unsichtbar, funktionell – sprich für die Berechnungen, ist dieser Bereich aber nach wie vor vorhanden.

Im obigen Beispiel aus Abschnitt 8 soll die Spalte D, die lediglich für interne Berechnungszwecke verwendet wird, ausgeblendet werden. Dazu wird ein Rechtklick auf den Spaltenkopf vollzogen und im Kontextmenü der Befehl „Ausblenden“ verwendet.



	A	B	C	E	F	G	H	I
1	Aufnahme	Entlassung	Geschlecht	Dauer				
2	01.01.2001	03.01.2001	m	2		Männer :	50,00	
3	01.01.2001	02.01.2001	m	1		Frauen :	50,00	
4	03.01.2001	04.04.2001	w	91				
5	05.01.2001		m			Mittelwert :	26,75	
6	06.01.2001	06.01.2001	w	1		MW Mä :	17,75	
7	08.01.2001	12.03.2001	m	63		MW Fr.	35,75	
8	09.01.2001	11.01.2001	w	2				
9	09.01.2001	14.01.2001	m	5				
10	10.01.2001		w					
11	11.01.2001	01.03.2001	w	49				

Wie auf der obigen Abbildung zu erkennen ist, fehlt zwischen den sichtbaren Spalten C und E die ausgeblendete Spalte D.

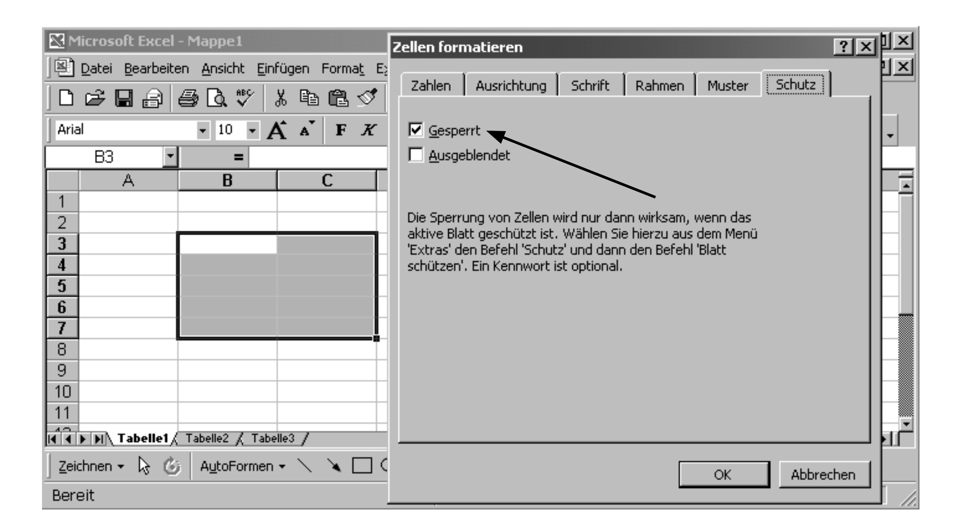
Um eine Spalte wieder einzublenden, markieren Sie den Bereich auf dem Spaltekopf C und E gefolgt von einem Rechtsklick. Im Kontextmenü führt der Befehl „einblenden“ dazu, dass alle in diesem Bereich ausgeblendeten Spalten wieder sichtbar werden.

09.2 Formeln, Zellen und Tabellen schützen

Microsoft Excel bietet die Möglichkeit gesamte Tabellen, aber auch Zellbereiche vor Veränderungen zu schützen. Ohne aktivierten Schutz kann jeder Anwender der Tabelle alle Bereiche der Tabelle verändern. Wird der Blatt- oder Mappenschutz aktiviert (Befehlsfolge EXTRAS -> SCHUTZ) so sind in der Standardeinstellung alle Zellen geschützt. Sie haben die Möglichkeit die Freigabe einer Mappe, inklusiver aller Objekte und Szenarien, von der Eingabe eines Passwortes abhängig zu machen.

Es sei jedoch erwähnt das im Internet sehr viele kleine Programme gibt, die den Passwortschutz von Microsoft Office Programmen wieder „aushebeln“. Eine hundertprozentige Sicherheit gibt es mit dem Blattschutz also nicht. Zumindest bedarf es ein wenig krimineller Energie, um in eine geschützte Tabelle einzudringen.

Das komplette Schützen einer Mappe oder Tabelle macht jedoch nur dann Sinn, wenn dem Anwender die Bereiche geöffnet bleiben, um Eingaben in der Tabelle vorzunehmen. Sie müssen diese Tabellenteile von Hand vom Schutz zuvor befreien. Dies wird mit einem Rechtsklick auf dem gewünschten Bereich vorgenommen:



Wählen Sie im Kontextmenü „Zellen formatieren“ und anschließend das Register „Schutz“. Mit dem dortigen Optionsfeld, lässt sich der Schutz für den gewählten Bereich ausschalten.

09.3 Nur gültige Eingaben zulassen

Der Blattschutz sorgt zwar für eine Sicherheit der Funktionen, jedoch nicht für sichere Eingaben von Seiten des Anwenders. Um nur zulässige Eingaben in die Tabelle aufzunehmen hat Microsoft Excel eine spezielle Funktion. Diese Funktion soll auf die Tabelle aus dem Abschnitt 8 (Jahrbuch) angewendet werden.

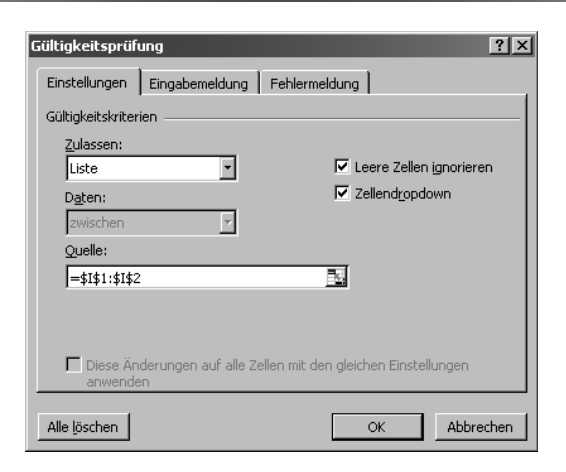
Die Spalten A und B dürfen nur Eingaben im Format Datum enthalten, um dies zu bewerkstelligen markieren Sie die beiden Spaltenköpfe und wählen Sie anschließend die Befehlsfolge DATEN -> GÜLTIGKEIT:



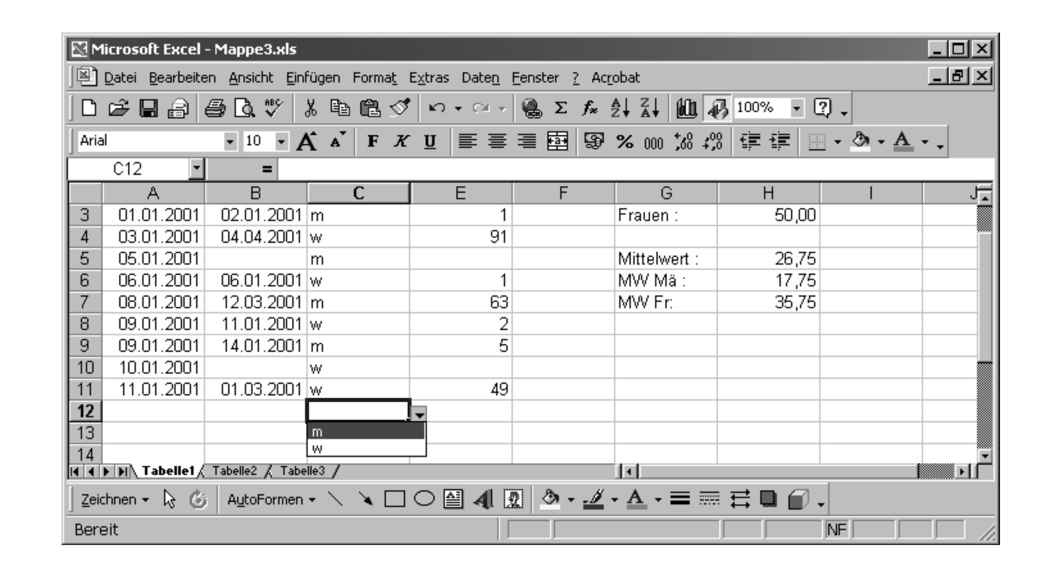
Im obigen Fenster lässt sich im Listenfeld „Zulassen“ das Format „Datum“ eintragen, weiter lässt sich der Bereich definieren. Somit ist möglich den Eingabebereich beispielsweise auf das laufende Jahr einzugrenzen.

Um für die Spalte C der Beispieltabelle lediglich für die Eingabe des Geschlechtsmerkmal „w“ und „m“ zu begrenzen, müssen die beiden zulässigen Eingaben zunächst in einem anderen Bereich der Tabelle hinterlegt werden. Schreiben Sie in die Zelle I1 ein m und in die Zelle I2 ein w.

Anschließend muss die Spalte C markiert werden und erneut die Befehlsfolge DATEN -> GÜLTIGKEIT verwendet werden. Im Gültigkeitsfenster ist nun im Listenfeld „Zulassen“ „Liste“ auszuwählen.

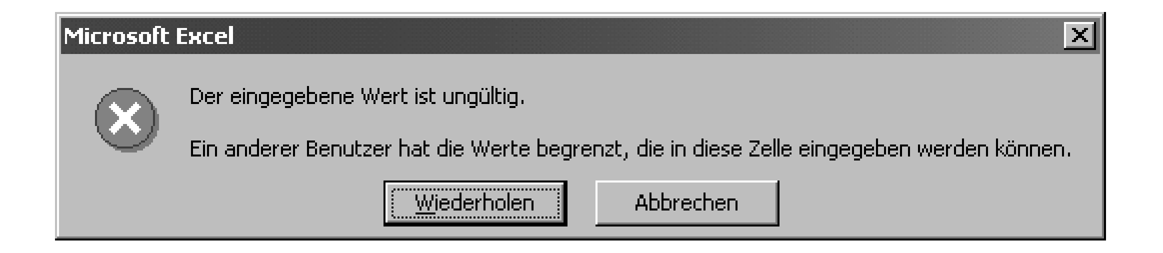


Als Quelle ist nun der Bereich I1 bis I2 auszuwählen. Wird nun der Anwender ein Geschlechtsmerkmal eingeben, so erscheint neben der Zelle ein kleiner Pfeil und die Werte „w“ und „m“ sind in der Auswahlliste zu finden. Eine Eingabe über die Tastatur ist jedoch nach wie vor ebenfalls möglich.



Ebenfalls im Fenster „Gültigkeit“ kann die Länge einer Texteingabe vorgegeben werden, soll beispielsweise eine feste Codierung, wie der medizinische ICD10 Code eingegeben werden, lässt sich über diese einfache Möglichkeit prüfen, ob die Eingabe überhaupt zulässig ist.

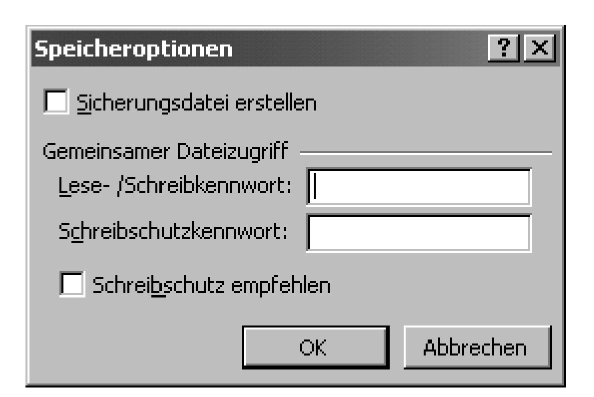
Versucht der Anwender einen nicht zulässigen Wert einzugeben, so erscheint ein Fehlerdialog von Microsoft Excel:



09.4 Nur Befugten das Öffnen ermöglichen...

Wie unter Microsoft Word, so haben Sie auch unter Excel die Möglichkeit beim Speichervorgang ein Passwort zu hinterlegen, welches das Öffnen der Datei von der korrekten Eingabe desgleichen abhängig macht.

Um bei der Speicherung ein Passwort zu hinterlegen wählen Sie den Dialog
DATEI -> SPEICHERN UNTER -> EXTRAS -> ALLGEMEINE OPTIONEN



Beachten Sie, dass Ihnen das Passwort anstelle des Klartextes nur in Sternchen angezeigt wird, das soll verhindern, dass Ihnen jemand über die Schulter schaut und das Passwort abliest. Zur Sicherheit müssen Sie nach Vergabe des Passwortes das gleiche Wort noch einmal eingeben.

Auch hier gilt wieder die Tatsache, dass es im Internet viele Programmanbieter gibt, die den Passwortschutz umgehen.

10 Excel-Befehle am Dienstplanbeispiel

Um die Dienstplanung mit Microsoft Excel durchführen zu können, sind eine Vielzahl verschiedener Befehle notwendig, die in diesem Kapitel vorgestellt werden. Allein mit den Befehlen die das Tabellenkalkulationsprogramm zu bieten hat, ist nur ein Teil der für die Dienstplanung notwendigen Funktionalitäten umzusetzen. In einem späteren Teil werden komplette Abläufe mit Visual Basic for Applications (VBA) realisiert. Oft sind Befehle sowohl in Excel als auch in VBA vorhanden. Welche von beiden Varianten man einsetzt ist reine Geschmackssache.

Zunächst muss der Plan in der Lage sein mit Buchstaben zu rechnen – der Dienstplan wird bekanntlich durch Buchstabensymbole dargestellt. Klassischerweise wird ein „F“ für einen Frühdienst verwendet, ein „S“ für einen Spätdienst. Am Bezirkskrankenhaus Kaufbeuren, welches bis in die 1990er Jahre nach einem sehr eigenwilligen Arbeitszeitmodell mit zwei Diensten (Tag- und Nachtdienst) mit je 12 Stunden arbeitete, wird der Dienst heute durch die Buchstaben A,B,C und W dargestellt. Dieses Benennungsschema findet sich auch in den Programmtiefen des fertigen Dienstplaners wieder – wobei die für die Planung eingesetzten Buchstaben flexibel zu wählen sind.

Im ersten Beispiel soll noch keine Flexibilität bei der Benennung möglich sein und es wird der F-, S- und W-Dienst abgebildet.

Microsoft Excel - Mappe1

	A	B	C	D	E	F	G
1							
2							
3	Mitarbeiter 1	S	F	S			
4	Mitarbeiter 2	F	S	S			
5	Mitarbeiter 3			F			
6	Mitarbeiter 4		W	W			
7							
8							
9							
10							
11							

Übernehmen Sie die Einträge aus der obigen Abbildung.

Es soll nun die Anzahl von „F“-Diensten für den jeweiligen Mitarbeiter errechnet werden. Der Befehl für die Zelle E3 lautet:

=ZÄHLENWENN(B3:D3;“F“)

Die Anführungszeichen im hinteren Bereich der Klammer sind notwendig um Excel mitzuteilen, es wird hier ein so genannter String – eine Zeichenkette – gesucht und keine Variable mit dem Namen F.

Die Formel lässt sich anschließend für die Mitarbeiter am Auffüllkästchen nach unten ziehen. Excel passt die entsprechende Formel für die jeweilige Zeile an.

Einmal angenommen, der Frühdienst sei 8 Stunden lang, ein Spätdienst 9 Stunden und eine Nachtwache 10 Stunden lang. Um nun aus der obigen Zeile einen Wert zu ermitteln muss die Formel entsprechend angepasst werden. Für die Zelle E3 lautet der Befehl:

=ZÄHLENWENN(B3:D3;"F")*8

Ändern Sie einfach mal einen Dienst von „S“ in „F“ und Sie werden das Ergebnis sofort in einem neuen Stundenwert sehen. Nun sollen die anderen zwei Dienstformen in die Berechnung mit einfließen. Der Befehl in E3 heißt entsprechend:

=ZÄHLENWENN(B3:D3;"F")*8+ZÄHLENWENN(B3:D3;"S")*9+ZÄHLENWENN(B3:D3;"W")*10

Da Punktrechnung vor Strichrechnung geht ist in diesem Fall das Setzen zusätzlicher Klammern nicht notwendig.

Microsoft Excel - Mappe1

D11 = 10

	A	B	C	D	E	F	G
1							
2							
3	Mitarbeiter 1	S	F	S	26		
4	Mitarbeiter 2	F	S	S	26		
5	Mitarbeiter 3			F	8		
6	Mitarbeiter 4		W	W	20		
7							
8							
9			F	8			
10			S	9			
11			W	10			

Tabelle1 / Tabelle2 / Tabelle3

Bereit

Eine Änderung der Dienstdauer hätte, bei aktuellem Beispiel zur Folge, dass innerhalb der Funktionen Änderungen vorzunehmen sind – das soll mit einer Individualisierung verhindert werden. Sowohl die Dienstbezeichnung, als auch die Dienstdauer – hier im Multiplikator hinterlegt – soll nicht innerhalb der Funktion definiert werden.

Übernehmen Sie die Einträge aus der obigen Abbildung. Die Funktion in E3 muss entsprechend verändert werden:

=ZÄHLENWENN(B3:D3;C9)*D9+ZÄHLENWENN(B3:D3;C10)*D10+ZÄHLENWENN(B3:D3;C11)*D11

Leider lässt sich eine solche Funktion nicht mit dem Auffüllkästchen nach unten ziehen, da auch die Verweise auf die definierenden Zellen mitbewegt werden würde. Durch Zuhilfenahme von Dollarzeichen lässt sich dieses Verhalten verändern:

=ZÄHLENWENN(B3:D3;$C$9)*$D$9+ZÄHLENWENN(B3:D3;$C$10)*$D$10+ZÄHLENWENN(B3:D3;$C$11)*$D$11

Jetzt lässt sich die Funktion auf die anderen Zellen „herunterziehen“. Hübsch ist jedoch etwas anderes und die Funktion lässt sich ohne ständigen Blick auf die Tabelle kaum lesen. Sprechender Namen wären hier angebracht – Variablennamen (vgl. Kapitel 5.1).

Geben Sie im Namenfeld folgende Bezeichnungen ein:

C9	:	Frueh
C10	:	Spaet
C11	:	Wache
D9	:	Frueh_h
D10	:	Spaet_h
D11	:	Wache_h

Bei der Vergabe von Variablennamen in der Programmierung empfiehlt sich die Beachtung folgender Regeln:

- Variablennamen sollen für sich sprechend sein
- es sollen keine Umlaute benutzt werden
- verwenden Sie niemals Leerzeichen im Variablennamen
- verwenden Sie das eigene Variablenschema möglichst durchgängig
- die Variablennamen nicht durchwegs groß schreiben

Variablennamen – in der Excel-Oberfläche spricht man von Zellnamen, haben eine hervorragende Eigenschaft. Sie gelten über die Tabellengrenze innerhalb eines Excel-Dokuments. Sprich: Alle Definitionen lassen sich auch in einer separaten Tabelle zusammenfassen.

Die Zelle E3 lässt sich nun entsprechend neu aufbauen:

=ZÄHLENWENN(B3:D3;Frueh)*Frueh_h+ZÄHLENWENN(B3:D3;Spaet)*Spaet_h+ZÄHLENWENN(B3:D3;Wache)*Wache_h

Auch diese Funktion lässt sich mit dem Auffüllkästchen problemlos nach unten „ziehen“. Da die Excel-Schlüsselwörter für die Befehle aus Großbuchstaben bestehen und die definierten Variablen nur mit einem Großbuchstaben beginnen, sind sie klar voneinander zu unterscheiden.

10.1 Umstellung auf Uhrzeitwerte

Die obige Tabelle ist nun in der Lage korrekt zu rechnen, nur werden anstelle der gewünschten Stunden- und Minutenwerte noch Dezimalzahlen ausgegeben. Diese Umstellung ist problemlos möglich – vertiefte Informationen zur Uhrzeitberechnung wurden in den vorhergehenden Abschnitten bereits genannt.

Microsoft Excel - Mappe1

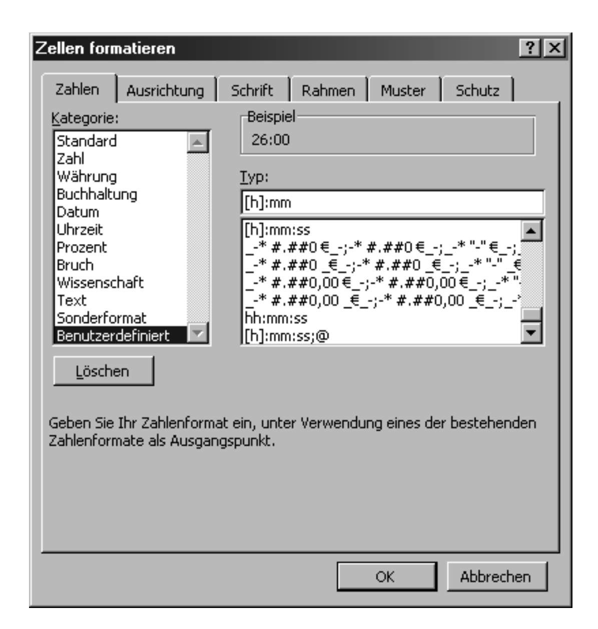
Spaet_h fx 9

	A	B	C	D	E	F	G
1							
2							
3	Mitarbeiter 1	S	F	S	18,3333333		
4	Mitarbeiter 2	F	S	S	18,3333333		
5	Mitarbeiter 3			F	0,33333333		
6	Mitarbeiter 4		W	W	20		
7							
8							
9			F	08:00			
10			S	9			
11			W	10			
12							

Tabelle1 / Tabelle2 / Tabelle3

Bereit

Wird der Stundenwert in D9 entsprechend der Reglements zur Eingabe von Uhrzeiten auf 08:00 geändert, so führt dies zunächst zu einem unhübschen Fehler in der Summenrechnung, da nun Dezimalberechnung und Uhrzeitrechnung aufeinandertrifft. Erst wenn alle Stundenwerte im Uhrzeitformat eingetragen wurden, sollte der Bereich E3 bis E6 in folgendes Format gebracht werden (Markieren -> Format -> Zellen):



Das benutzerdefinierte Format bietet eine flexible Anzahl von Stunden [h] an, der Minutenwert wird stets zweistellig angegeben.

10.2 Bereitstellung einer Soll/Ist Prüfung

Eine wichtige Funktion ist die Prüfung, ob im Dienstplan alle Dienste ausreichend besetzt wurden. Für diese Funktion werden „Bedingte Formatierungen" verwendet (siehe Abschnitt 7.2). Im DPE3 wird die Prüfung ein wenig komplexer, da der jeweilige Wochentag analysiert werden muss, im Abschnitt 10.9 findet sich eine Übersicht aller funktionellen Bereiche des DPE3. Mit dem Befehl =ZÄHLENWENN() wird die Anzahl der in einer Dienstform vorhandenen Dienste gezählt und mit einem Sollwert verglichen.

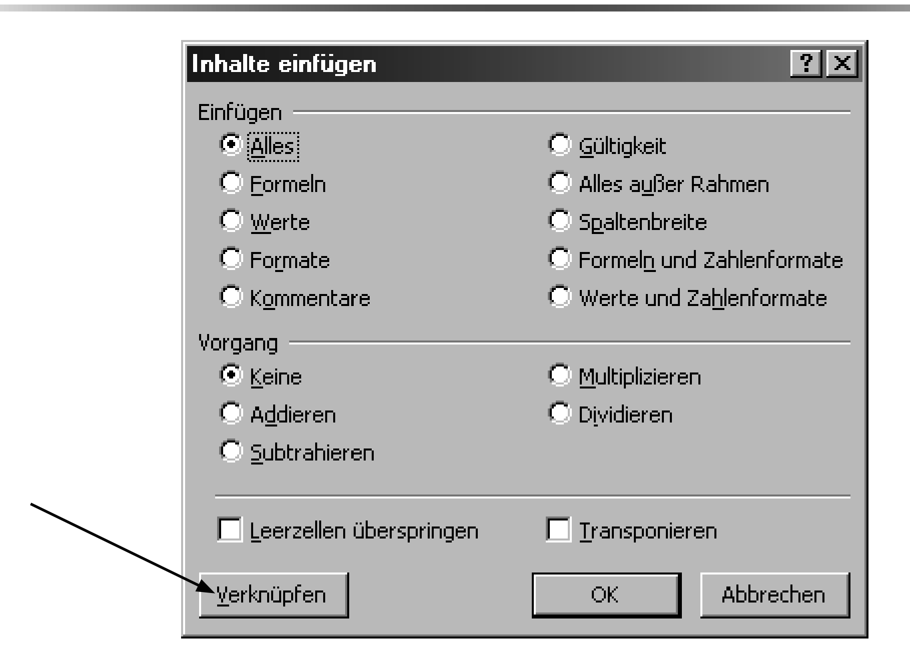
10.3 Arbeiten über die Tabellengrenze hinweg

Mit Hilfe der Vergabe von Zellnamen (Abschnitt 5.2) lässt sich der Zugriff auf eine Information auch über die Tabellengrenze hinweg realisieren. Im obigen Tabellenbeispiel würde sich der Umzug der Stundendienste in eine andere Tabelle empfehlen. Dazu wird der Bereich C9 bis D11 markiert und mit Bearbeiten -> Ausschneiden in die Zwischenablage abgelegt. Durch einen Klick auf die Registerleiste „Tabelle 2" wird die zweite Tabelle in den Vordergrund geholt und mit einem Klick auf Bearbeiten -> Einfügen werden die gewählten Zellen in die zweite Tabelle verschoben.

Innerhalb der Tabelle 1 wird sich in der Berechnung nichts verändern, da durch die Benennung der Zellen ein übergreifendes Arbeiten ermöglicht wird. Es lässt sich jedoch auch ohne Vergabe eines Zellnamens der Zugriff auf eine andere Tabelle ermöglichen:

```
=Tabelle2!$B$4
```

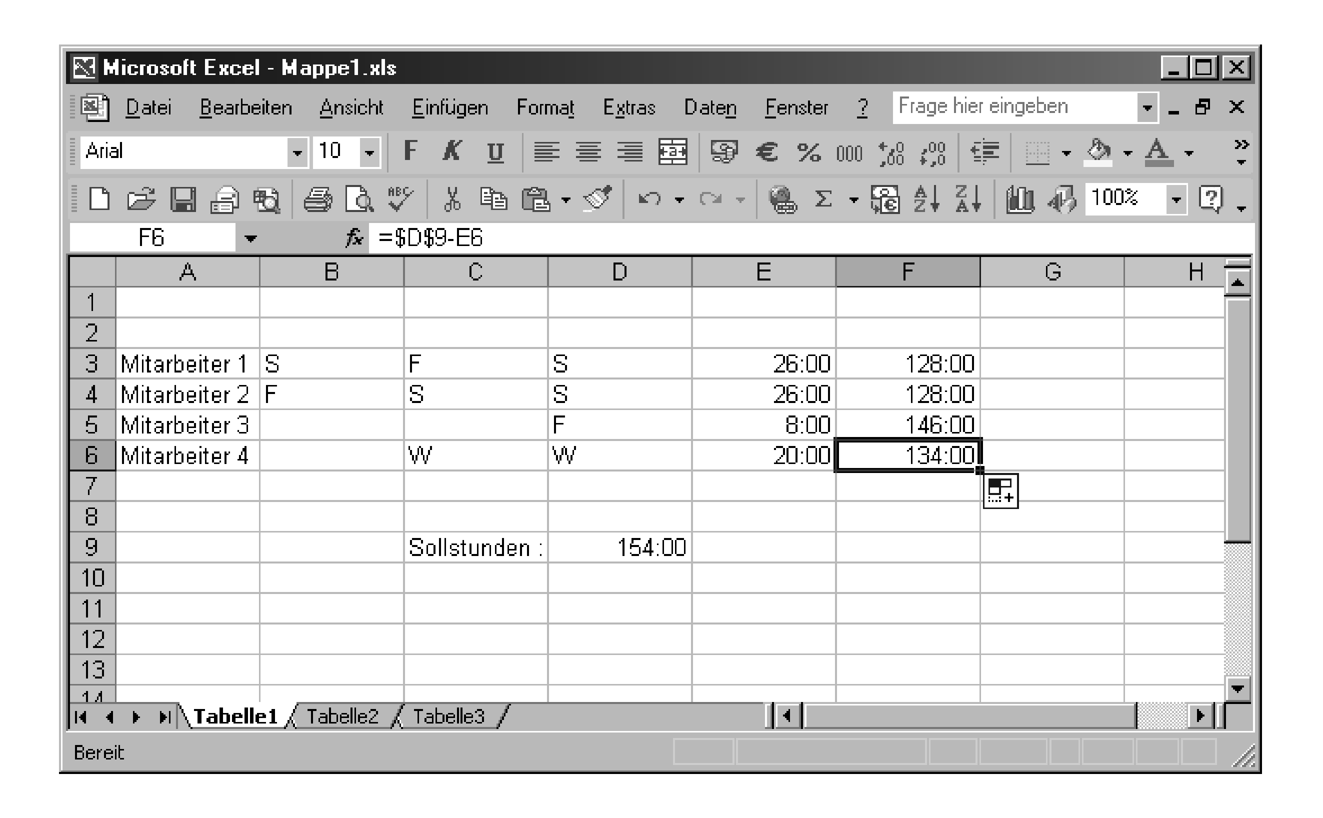
Die einfachste Methode einen Bereich von einer Tabelle mit einer Region einer anderen Tabelle kommunizieren zu lassen ist über die Zwischenablage. Kopiert man einen Tabellenbereich in die Zwischenablage empfiehlt es sich nicht einfach mit EINFÜGEN im Menü Bearbeiten das Einfügen zu bewerkstelligen, sondern über INHALTE EINFÜGEN mit der Option „Verknüpfen".



10.4 Sollstunden

Das Berechnen von Sollstunden geschieht über die Methoden, die bereits in der allgemeinen Einführung zu Microsoft Excel genannt wurden. Als Zellformat empfiehlt sich die benutzerdefinierte Auswahl [h]:mm, da nur in diesem Format die Darstellung von Stunden im Hunderter-Bereich korrekt umgesetzt wird.

Für das obige Beispiel ergibt sich folgende Darstellung:



10.5 Teile von Textinhalten

Oft ist es in der Praxis notwendig in einer Zelle mehr als nur eine Information abzulegen. Sowohl Excel selbst, als auch VBA bieten Möglichkeiten nur einen Teil einer Zellinformation auszulesen. Genutzt wird diese Methode beispielsweise bei der Definition der Mitarbeiter:

	A	B
9	**Muster D.**	**00:00**
10	SL - WT	
11	390	

Die mittlere Zelle enthält sowohl die Dienstbezeichnung des Mitarbeiters, hier Stationsleiter (SL) und, hinter dem Gedankenstrich, die Angabe zur Anstellung, hier Wochenturnus. Um aus dieser Zelle die gewünschte Information einzeln auszulesen bieten sich die Befehle LINKS und RECHTS an.

= LINKS(A10;3)

liefert als Ergebnis SL_ (der Unterstrich steht hier für ein Leerzeichen)

= RECHTS(A10;3)

liefert als Ergebnis _WT (der Unterstricht steht hier für ein Leerzeichen).

Beide Befehle werden im DPE3 sehr oft verwendet, um aus den für den menschlichen Benutzer besser lesbaren Buchstaben Berechnungen aufzubauen.

Soll ein Textteil aus der Mitte ausgelesen werden, so wird der Befehl TEIL verwendet:

=TEIL(A10;5;1)

Liefert als Ergebnis den Gedankenstrich. Nach der Angabe für den Text (A10), wird die Position des ersten Zeichens von links betrachtet aus benannt (5) und anschließend die Anzahl von Zeichen (1). Unter VBA steht ebenfalls dieser Befehl zur Verfügung und wir MID$ genannt.

10.6 Versteckte Bereiche des DPE3

Um alle Bereiche des Dienstplans einsehen zu können ist zunächst der Blattschutz zu deaktivieren. Anschließend wird die gesamte Spalte AL markiert und in Richtung grauer Rand gezogen. Mit einem Rechtsklick auf den Spaltenkopf AL erscheint der Befehl „Einblenden" über den die versteckten Spalten im Plan sichtbar gemacht werden können.

10.6.1 Korrektes Monatsende erfassen

Der DPE3 bildet stets 31 Tage für den Planungsmonat ab. Um zu verhindern, dass versehentlich in die ersten Tage des Folgemonats geplant wird, prüft das VBA-Makro „Plan-Killer" in Verbindung mit dem Bereich AO3 – AS3 ob die letzten Tage eines Monats überhaupt zulässig sind.

SUM	28	29	30	31
0	0	0	0	0

=MONAT(AE5)-MONAT($I$1)

Die hier gewählte Funktion aus AP3 liest sich wie folgt: „Ziehe von dem Monat, den du in AE5 (der 28te Tag) findest den Planmonat ab". Diese Rechnung ergibt in der Regel das Ergebnis 0. Sollte beispielsweise der Monat nur 30 Tage haben, so wird die Funktion in AS3 das Ergebnis 1 erzeugen. In AO3 wird die Summe aus dem Bereich gebildet, um für andere Programmstellen im Plan die Anzahl von Tagen, die dem laufenden Plan auf 31 fehlen zu repräsentieren.

10.6.2 Teamsitzungsmarker

In der Zelle AR4 wird, sofern keine Teamsitzung im laufenden Monat geplant wurde, eine Null gespeichert. Nachdem das VBA-Makro zu Erstellung der Teamsitzung durchgelaufen ist, wird hier der Wert auf 1 korrigiert. Damit soll verhindert werden, dass ständig der Makrodialog eingeblendet wird, sobald der Benutzer einen Eintrag in der Nähe des Schlüsselwort für die Teamsitzung durchführt. Mit dem Befehl „Teamsitzungsmarker zurücksetzten" aus dem Menü „DIV" wird dieser Marker wieder auf Null zurückgesetzt. Somit ist das erneute Aufrufen des Makros möglich.

10.6.3 Dienstberechner

Im Bereich AM8 bis AV72 wird die Berechnung der Dienstzeit vorgenommen. Kern der Berechnung ist die Feststellung, welchen Dienst hat der Mitarbeiter wie häufig im Plan gearbeitet.

Die Funktion aus AM9 soll hier als Beispiel dienen:

=WENN(AV9=0;ZÄHLENWENN(D9:AH9;LINKS(BEZ_A;P_ZEL))*S_A;ZÄHLENWENN(D9:AH9;LINKS(BEZ_A;P_ZEL))*WT_A)

Die erste Wenn-Prüfung fragt ab, ob in AV9 eine Null angezeigt wird. Wenn dies der Fall ist, wird der gesamte Planungsbereich D9 bis AH9 auf den Bezeichner BEZ_A (typischer Frühdienst „A") abgefragt. Der Befehl LINKS bestimmt in Bezug auf den Parameter aus der Tabelle DEFINITIONEN wie viele Stellen als gültig angesehen werden sollen. Die Summe aus dem ZÄHLENWENN wird mit dem Stundenwert für den ersten Frühdienst S_A multipliziert. Wird in AV9 ein anderer Wert als 0 gefunden, so wird die gleiche Berechnung durchgeführt, nur mit dem Stundenwertmultiplikator für den Wochenturnus.

Die folgenden Spalten beinhalten jeweils den gleichen Befehl, nur für unterschiedliche Dienstformen. Eine Besonderheit stellt nun der Urlaub- und geplante Krankheitsdienst dar, da dieser entsprechender der Anstellung (Voll- oder Teilzeit) nur entsprechend angerechnet wird. Darum findet sich in der Spalte AS folgende Funktion:

=WENN(RECHTS(A10;3)="1/1";1;0)+WENN(RECHTS(A10;3)="1/2";0,5;0)+WENN(RECHTS(A10;3)="3/4";0,75;0)+WENN(RECHTS(A10;3)="1/4";0,25;0)+WENN(RECHTS(A10;3)=" WT";1*WTVZ;0)

Die Spalte AS bildet einen abschließenden Multiplikator für Urlaubs- und Krankheitstage, der insgesamt fünf verschiedene Werte annehmen kann:

Dienst	Wert
1/1	1
3/4	0,75
1/2	0,5
1/4	0,25
WT	WTV Z***

*** = Wochendienst VZ Multiplikator aus der Tabelle DEFINITIONEN, in der Funktion als Zellname WTVZ

In der Spalte AV wird lediglich geprüft, ob der jeweilige Mitarbeiter im Wochenturnus angestellt ist (Merkmal: „ WT") oder nicht. Konzeptionell bieten sich hier Verbesserungsmöglichkeiten, falls mehr als ein Mitarbeiter der Station im Wochenturnus beschäftigt ist. Ist dies der Fall, ist nur zu hoffen, dass beide Mitarbeiter die gleiche Sollstellung haben. Ein abweichende Berechnung der Sollstunden für Wochenturnusmitarbeiter steht bereits auf der Wunschliste für die geplante Version 3.5 des Dienstplan.

10.6.4 Speicherort für Wünsche

Die Spalte AW wird als Speicherort für die Mitarbeiterwünsche genutzt. Über die Dialogfelder, die mit Hilfe von VBA angesteuert werden, wird dieser Bereich mit Eingaben und Änderungen versorgt. Das Verfahren hinter den Wünschen ist relativ einfach:

200,310,420,530,946

Ein solcher Eintrag liest sich wie folgt

Wunsch	Wert	Inhalt
1	200	Montag gar keinen Dienst.
2	310	Dienstag ungern Dienst.
3	420	Mittwoch gern Dienst.
4	530	Donnerstag am liebsten Dienst.
5	946	Jeden Tag ungern Nachtwache.

Jeder der fünf möglichen Wünsche besteht aus einem dreistelligen Code:

Stelle	Wert	Inhalt
1	1	Sonntag
1	2	Montag
1	3	Dienstag
1	4	Mittwoch
1	5	Donnerstag
1	6	Freitag
1	7	Samstag
1	8	Feiertag
1	9	Jeden Tag

Stelle	Wert	Inhalt
2	0	gar keinen Dienst.
2	1	ungern Dienst.
2	2	gern Dienst.
2	3	am liebsten Dienst.
2	4	ungern ***
2	5	gern ***

*** = Dienstform von der dritten Stelle

Stelle	Wert	Inhalt
3	1	DEFINITION C18
3	2	DEFINITION C19
3	3	DEFINITION C20
3	4	DEFINITION C21
3	5	DEFINITION C22
3	6	DEFINITION C23

10.6.5 Nachtwachenspeicher

Die Spalte AX wird für die Speicherung der Anzahl von Nachtdiensten im laufenden Quartal benutzt. Wird ein Planmonat innerhalb des laufenden Quartals eröffnet, so werden durch das VBA-Makro „Plan Killer“ die gefunden Nachtdienste dem Wert in AX hinzugefügt. Ist die Erstellung eines neuen Planungsmonat außerhalb des aktuellen Quartals, so wird dieser Speicherort wieder auf 0 zurückgesetzt. Wann ein Quartalsabschnittwechsel vorliegt wird aktuell durch den Wert 3 in der Spalte E der Tabelle JAHRESDATEN entnommen.

Soll anstelle eines Quartals-Berechnens ein Jahres-Nachtdienst-Zähler eingerichtet werden, so ist jedem Monat des laufenden Jahr in der Spalte E der Tabelle JAHRESDATEN mit Ausnahme des Januar ein Wert abweichend von 3 zuzuordnen.

10.6.6 Spiegel / Schattenplan

Der gesamte Dienstplan aus dem Planungsbereich findet sich in einer Kopie im Bereich BN10 bis CR73. Hier werden einerseits die Einträge für den Mitarbeiterplan abgespeichert, andererseits wird hier die Soll/Ist-Besetzung durchgerechnet. Im Spiegelplan wird der gewünschte Dienst auf die gewünschte Länge getrimmt, die in der Tabelle DEFINITIONEN als Parameter festegelegt wurde.

Eine besondere Funktion hat die Spalte BM. Hier soll die Funktion für den ersten Mitarbeiter stellvertretend vorgestellt werden:

```
=WENN(LINKS(A10;3)="SL„;AS_SL;0)+WENN(LINKS(A10;3)="SLV";AS_SLV;0)+
WENN(LINKS(A10;3)="KS „;AS_KS;0)+ WENN(LINKS(A10;3)="KP „;AS_KP;0)+
WENN(LINKS(A10;3)="FS „;AS_FS;0)+ WENN(LINKS(A10;3)="FP „;AS_FP;0)+
WENN(LINKS(A13)="HEP";AS_HEP;0)+ WENN(LINKS(A10;3)="KPH";AS_KPH;0)+
WENN(LINKS(A10;3)="AS „;AS_AS;0)+ WENN(LINKS(A10;3)="AP „;AS_AP;0)+
WENN(LINKS(A13)="ZDL";AS_ZDL;0)+ WENN(LINKS(A10;3)="KPS";AS_KPS;0)+
WENN(LINKS(A0;3)="SON";AS_SON;0)+ WENN(LINKS(A10;3)="FPK";AS_KPK;0)+
WENN(LINKS(A10;2)="SD";AS_SD;0)
```

Dieser, zugegeben etwas lang geratene Befehl, stellt fest, ob der Mitarbeiter in die Sollbesetzung mitgerechnet werden soll, oder nicht. Wie sich erkennen lässt, ist eine Flexibilisierung an dieser Stelle einfach umzusetzen. Der jeweilige Parameter beginnend mit AS_ wird über die Tabelle DEFINITIONEN parametriert.

Der Bereich BN77 bis CR82 zählt lediglich die sechs möglichen Dienstformen mit einem ZÄHLENWENN Befehl.

Im Bereich BN83 bis CR84 wird analysiert, welcher Wochentag vorliegt. So liest sich Funktion aus BN83:

```
=WENN(D6=3;1;0)+WENN(D6=4;1;0)+WENN(D6=5;1;0)+WENN(D6=6;1;0)+WENN(D6=7;1;0)
```

Ist der erste Planungstag ein Montag (3), Dienstag (4) … Freitag(7). Microsoft Excel zählt die Wochentag in unterschiedlicher Art und Weise, als es VBA tut – dies ist zu beachten.

Die Zelle BN84 sucht erwartungsgemäß nach Samstagen und Sonntagen:

```
=WENN(WENN(D6=1;1;0)+WENN(D6=2;1;0)+WENN(D7="FT";1;0)>1;1;WENN(D6=1;1;0)+
WENN(D6=2;1;0)+WENN(D7="FT";1;0))
```

aber auch nach dem Schlüsselwert „FT“ in der Zeile unterhalb des Datums. Es wird somit unterschieden ob ein Wochentag oder ein Wochenende beziehungsweise Feiertag vorliegt. Diese Vorgehensweise geht auf die Version 2.x des Dienstplan zurück, bei der eine einzelne Sollstellung nicht möglich war. Mit der Version III wurde die Möglichkeit eingeführt jeden einzelnen Wochentag unterschiedlich in die Sollstellung zu bringen, um beispielsweise zusätzliches Personal für eine Visite an einem Dienstag einplanen zu können. Welches Verfahren eingesetzt wird, wird über den Parameter „PS“ Plansoll Zelle C75 in der Tabelle DEFINITIONEN festgelegt.

Es gibt somit zwei unterschiedliche Sollstellungsbereiche in DPE3, einen für das neuere Wochentag-Basierende Verfahren (PS=0) und eines für das ältere Verfahren ohne Wochentagaufschlüsselung (PS=1). Der Bereich BN96 bis CR100 bildet die „Effektive Sollstellung“. Dieser Bereich bildet auch die Grundlage für den „farbigen Block“ unterhalb des Dienstplaners, mit dessen Hilfe festgestellt wird, ob alle Dienste ausreichend mit Personal versorgt werden.

Die Funktion aus BN96:

=WENN(BN83=BN84;BN91;BN86+BN91)+BN102

Ist der Wochentag gleich einem Wochenendtag (Feiertag) dann wählte die Sollstellung aus dem Wochenendtag, ansonsten addiere die Sollstellung aus dem regulären Tag mit dem Wochenendtag. Anschließend wird der Wert aus dem wochentagbasierenden stets addiert. Da die wochentagbasierende Berechnung und das ältere Verfahren einen Multiplikator, beziehungsweise einen umgekehrten Multiplikator auf den Parameter (Zellname) PS besitzen, wird immer nur eines der Verfahren eingesetzt.

Die Bereitstellung für einen regulären Dienst (BN86):

=WENN(BN83=1;SOLL_FR*PS;0)

Findet sich ein nicht Wochenende/Feiertag so wird die Sollstellung für den Frühdienst (SOLL_FR) mit dem Parameter PS (0/1) multipliziert. Findet sich kein normaler Werktag nimmt die Zelle den Wert 0 an.

Die Bereitstellung für einen Wochentag-Dienst (BN102):

=(NICHT(PS)*(WENN(WOCHENTAG(D5)=1;D_A_1;0)+WENN(WOCHENTAG(D5)=2;D_A_2;0)+
WENN(WOCHENTAG(D5)=3;D_A_3;0)+WENN(WOCHENTAG(D5)=4;

D_A_4;0)+WENN(WOCHENTAG(D5)=5;D_A_5;0)+WENN(WOCHENTAG(D5)=6;D_A_6;0)+
WENN(WOCHENTAG(D5)=7;D_A_7;0)))*NICHT(D7=“ft“)+WENN(D7=“ft“;
D_A_F;0)*NICHT(PS)

Neben der Aufschlüsselung des Wochentags zur jeweiligen Sollstellung (D_A_1 = Frühdienst „A“ für einen Samstag, D_A_2 = Frühdienst „A“ für einen Sonntag …) wird mit einer umgekehrten Logik (Befehl NICHT) dafür gesorgt, dass diese Berechnung nur dann durchgeführt wird, sofern der Parameter PS auf 0 gesetzt ist.

10.6.7 Minusübergabe

Die Spalte CT wird lediglich als Datenspeicher für eine Berechnung genutzt, die es unter Microsoft Excel normalerweise nicht gibt. Zwar ist Microsoft Excel in der Lage bei Einschaltung des 1904er Jahresformat (siehe Vorbereitungen) negative Uhrzeit- und Datumswerte auszugeben, nicht jedoch die Eingabe negativer Werte entgegen zu nehmen. Diese Werte werden jedoch für die Stundenübernahme aus dem Vormonat benötigt, falls ein Mitarbeiter beispielsweise mit -23:12 Stunden in den neuen Monat startet. Die Eingaben werden in der ersten Zeile in Spalte B bei einem jeden Mitarbeiter entgegen genommen. Durch „Worksheet_SelectionChange“ im VBA-Code für Tabelle 1 (Plan) werden die Eingaben in dem Textfeld durch eine Multiplikation mit -1 umgekehrt und in Spalte AK zu den Daten des aktuellen Plans addiert.

VBA – Code zur Vorzeichenumkehr :

```
‘  Wurde eine Änderung außerhalb Spalte ‚B‘ durchgeführt ?
If akt_col <> 2 Then GoTo pruefung2
If akt_row < 9 Or akt_row > 73 Then Exit Sub
If akt_row Mod 3 <> 1 Then Exit Sub
‘  Wenn der Programmverlauf hier noch nicht weitergesprungen ist, so wurde eine
   Änderung beim STUNDENKONTO-Übertrag durchgeführt.
‘  Warum ist dies besonders hervorzuheben ?
‘  Die Eingabe von negativen STUNDEN:MINUTEN Werten ist mit Excel eigentlich nicht möglich.
```



```
‘  Mit dem Multiplikator -1 und dem Rechnen auf Basis von 1904
   (Extras -> Optionen -> Berechnen) ist dies jedoch mit VBA möglich.
wert = Cells(akt_row - 1, 2)
If wert = „“ Then wert = 0
If Left$(wert, 1) <> “-” Then
Cells(akt_row - 1, 98).Value = wert
Else
Cells(akt_row - 1, 99).Value = Right$(wert, Len(wert) - 1)
Cells(akt_row - 1, 98).Value = Cells(akt_row - 1, 99).Value * -1
Cells(akt_row - 1, 99) = 0
End If
Exit Sub
```

10.6.8 Gültigkeitsprüfung

Im versteckten Bereich der Tabelle AX2 bis AX3 wird der Minimal- und Maximalwert für die Eingabe von Plusstunden in der Spalte AI zwischengespeichert. Die Werte werden eigentlich über die DEFINITIONEN-Tabelle festgelegt, jedoch ist bei der Gültigkeitsprüfung von Microsoft Excel kein Verweis auf eine andere Tabelle zulässig. Aus diesem Grunde wurden die beiden Werte hier als Verweis auf Zellenebene „zwischenspeichert“.

11 VBA in Excel nutzen

Das Programmieren in Microsoft Office hat sich mit den Jahren stark verwandelt. Ging es ursprünglich darum lediglich sich wiederholende Befehle zu automatisieren, um beispielsweise in Microsoft Word immer wieder die gleichen Formatbedingungen zu schaffen, so steht heute mit „Visual Basic for Applications“ (kurz **VBA**) eine mächtige Sprache zur Verfügung, die an Unübersichtlichkeit kaum zu überbieten sein dürfte. Selbst gestandene Excel-Experten finden für das ein und dasselbe Problem mehrere Lösungswege. Einige Befehlsstrukturen sind aus Kompatiblitätsgründen zu Vorgängerversion immer noch anwendbar, sodass neue und alte Programmiermethode gleichzeitig eingesetzt werden kann.

Im Gegensatz zur Programmierumgebung „Visual Basic 6“, der VBA in großen Teilen sehr ähnlich ist, wird der VBA-Code nicht in **Maschinensprache** übersetzt. Maschinensprache ist die Programmiersprache, die direkt vom Prozessor und seiner Architektur verstanden wird und somit auch in voller Geschwindigkeit ausgeführt werden kann. In der Praxis wird in den seltensten Fällen direkt auf dieser „niedrigen Ebene“ der Programmierung gearbeitet, da es sehr unkomfortabel ist, sich alle Spezifika der Prozessoren zu merken. Aus diesem Grunde wurden die Hochsprachen eingeführt, die eine Programmierung auf einer höheren Abstraktionsebene zulassen und gleichzeitig eine für den Menschen viel einfacher zu erlernenden Syntax mitbringen. VBA, welches für die Zusammenarbeit mit Office optimiert ist, verfügt zwar über den Befehl „kompilieren“, jedoch wird anstelle von Maschinencode ein Pseudocode erzeugt, der zügig interpretiert werden kann.

Anstelle sich beispielsweise mit der Funktion der grafischen Ausgabe „herumplagen“ zu müssen, wird in Visual Basic einfach gesagt: „Ich möchte hier ein Dialogfenster, in dem steht „das und das“, dieses Fenster kann mit JA, NEIN oder ABBRECHEN verlassen werden.“ Ob der Computer nun mit einem älteren Microsoft Windows 95 oder mit einer aktuelleren Betriebssystemversion betrieben wird, welche Prozessor das Herz der Maschine darstellt und ob der Monitor 17“ Röhre oder 19“ Flat-Screen ist, spielt für den Programmierer in den **Hochsprachen** kaum eine Rolle.

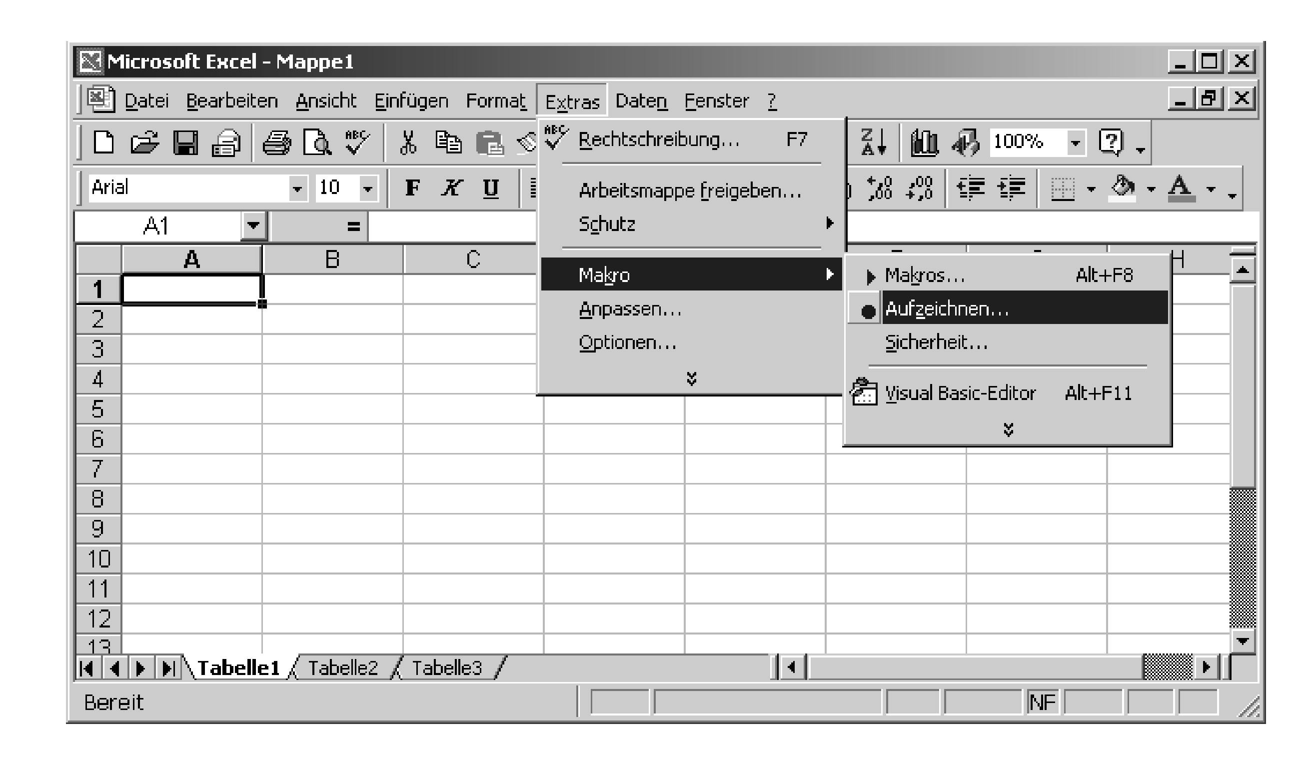
Visual Basic – die echte Programmierumbegung, bietet die Möglichkeit, bei Abschluss der Programmierarbeit aus dem Programm eine „echte“ ausführbare Datei für die Windows-Umgebung zu erzeugen, eine so genannte EXE-Datei. Mit welcher Sprache diese Datei erzeugt wurde merkt der normale Anwender nicht mehr. Auch ist es kaum möglich die EXE-Datei wieder in ihre Ursprungsversion zurückzuverwandeln. Das Visual-Basic Programm ist **kompiliert** worden – zu einer vom Betriebssystem lesbaren Form der Maschinensprache. Zwischen den verschiedenen Hochsprachen, beispielsweise C++, Visual Basic oder die modernenen Java-Sprache gibt es diverse Unterschiede, die hier nicht betrachtet werden sollen. Visual Basic ist eine der einfachsten Programmiersprachen die es gibt – zudem bietet die Sprache brauchbare Ergebnisse in kürzester Zeit.

Die Einführung in VBA wird hier bewusst **kurz gehalten**, und auf die Methoden beschränkt, die für das Verständnis von DPE3 erforderlich sind. Für einen tieferen Einstieg empfiehlt sich das Buch „Excel programmieren“ aus dem Addison Wesley Verlag von Michael Kofler – einer wahren Koriphäe auf dem Gebiet Visual Basic.

11.1 Makrorekorder

Der einfachste Weg mit VBA in Kontakt zu treten ist immer noch der Makrorekorder. Die Funktion, die vom Benutzer durchgeführte Befehle quasi **mitnotiert** und später, an einer anderen Stelle wieder abspult. Das was der Makrorekorder mitnotiert ist nichts anderes als Quelltext für VBA. Der so entstehende Quelltext ist sicherlich nicht besonders platzsparend, dafür jedoch gut zu lesen. Ein gewisser Teil des „Dienstplan in Excel 3“ wurde auch mit dem Makrorekorder aufgezeichnet – besonders im Bereich „Neuer Plan erstellen“. Das komplette Entfernen der Dienste aus dem alten Monat oder das Entfernen von Formateinstellungen wurde mit dem Makrorekorder aufgezeichnet. Eine Optimierung dieses Quellcodeabschnitts ist sicherlich möglich, aber es gibt in der IT-Branche eine sehr alte Weisheit, die besagt: „Never change a running system“. Ein einmal funktionierendes System sollte nie ohne besonderen Grund verändert werden.

Ein erster Versuch, soll die Funktionsweise des Makrorekorders näher bringen :



Im darauf folgenden Dialogfeld kann ein Name spezifiziert werden und ein Kommentar zum Makro hinterlegt werden.

Makro aufzeichnen
Makroname:
Makro1
Tastenkombination: Strg+
Makro speichern in: Diese Arbeitsmappe
Beschreibung:
Makro am 08.02.2004 von Thomas Bär aufgezeichnet
OK Abbrechen

Soll die Ansammlung von Befehlen über die Tastatur direkt aufrufbar sein, so lässt sich hier eine **Tastenkombination** mit der Steuerungstaste festlegen. In der Standardeinstellung wird das Makro in die aktuelle Arbeitsmappe integriert, so ist gewährleistet, dass die Funktion auch dann zur Verfügung steht, wenn die Datei per Datenträger oder Netzwerk auf einen anderen Rechner übertragen wurde.

Während der Aufzeichnung findet sich in der Statusleiste von Microsoft Excel der Hinweis „Aufzeichn.“. Je nach Version erscheint ein kleines Fenster mit den Befehlen „Stopp“ und „Pause“ wie es vom handelsüblichen Kassettenabspielgerät her bekannt ist.

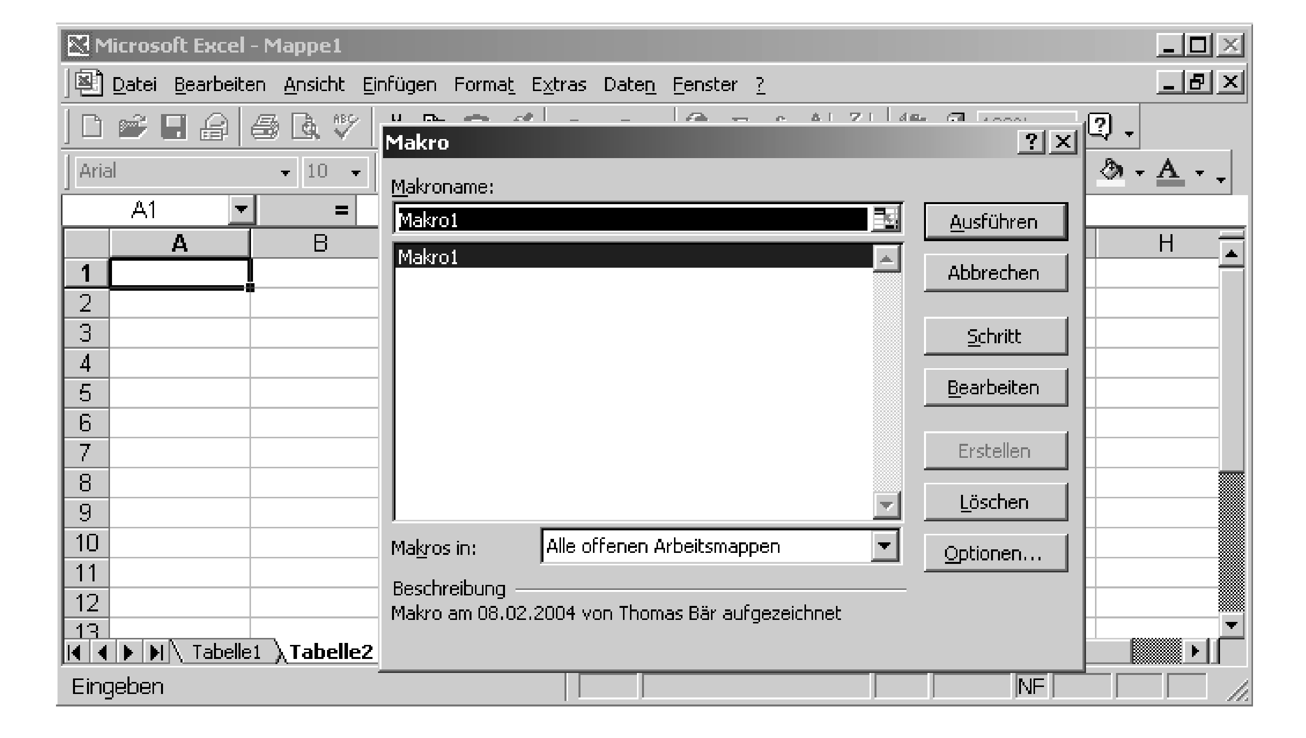
Während die Aufnahme aktiviert ist, wird jeder Befehl, den der Benutzer ausführt und jede Eingabe im Quellcode festgehalten. Wer üblicherweise viel mit der rechten Maustaste arbeitet, wird etwas enttäuscht: die **rechte Maustaste** ist während der Makro-Aufnahme-Phase gesperrt. Auch werden einige wenige Befehle aus den Menüleisten nicht mitprotokolliert. Eine genaue Erprobung des mitgeschriebenen Makros ist stets erforderlich.

Folgende Befehle werden nun während der Aufzeichnungs-Phase durchgeführt:

Zelle A1 wird mit dem Wert 120 versehen.
Zelle B1 wird mit dem Wert 130 versehen.
Zelle C1 wird mit der Funktion = A1+B1 versehen.
Zelle D1 erhält den Texteintrag „Hallo Welt“ (ohne Anführungszeichen).
Nach der Eingabe soll die Textfarbe der Zelle D1 noch auf ROT gesetzt werden.

Sofern das kleine Fensterchen mit dem „Stopp“ Befehl nicht angezeigt wird, kann die Erstellung des Makros mit dem Befehl Extras -> Makro -> Aufzeichnung Beenden abgeschlossen werden.

Wechseln man nun über die Register der Tabelle auf die „Tabelle 2“ und wählt anschließend den Befehl Extras -> Makro -> Makros so erscheint das Menü mit den bereits vorhandenen Makros zur Auswahl.



Wird das soeben aufgezeichnete Makro ausgewählt und die Schaltfläche „Ausführen“ angeklickt, so werden die Befehle in der gleichen Reihenfolge, wie in der Aufzeichnung auch auf diese zweite Tabelle angewendet.

Jetzt soll der Quelltext, den der Makrorekorder angelegt hat, einmal betrachtet und analysiert werden. Um die Anzeige des Quelltext zu erreichen, ist folgender Befehl notwendig :
Extras -> Makro -> Visual Basic Editor


```
Microsoft Visual Basic - Mappe1 - [Modul1 (Code)]

Sub Makro1()
'
' Makro1 Makro
' Makro am 08.02.2004 von Thomas Bär aufgezeichnet
'

'
    ActiveCell.FormulaR1C1 = "120"
    Range("B1").Select
    ActiveCell.FormulaR1C1 = "130"
    Range("C1").Select
    ActiveCell.FormulaR1C1 = "=RC[-2]+RC[-1]"
    Range("D1").Select
    ActiveCell.FormulaR1C1 = "Hallo Welt"
    Range("D1").Select
    Selection.Font.ColorIndex = 3
End Sub
```

Öffnen Sie auf der linken Fensterhälfte in der Explorer-Auflistung das Plussymbol bei „Module“ und das Modul1 – das soeben aufgezeichnete Makro – wird angezeigt. Wird es mit der linken Maustaste einmal angeklickt, so erscheint in der rechten Fensterseite der Visual Basic Quellcode des Makro.

```
01      Sub   Makro1()
02      ‘
03      ‘       Makro1 Makro
04      ‘       Makro am 08.02.2004 von Thomas Bär aufgezeichnet
05      ‘
06
07      ‘
08      ActiveCell.FormulaR1C1 = “120”
09      Range(“B1”).Select
10      ActiveCell.FormulaR1C1 = “130”
11      Range(“C1”).Select
12      ActiveCell.FormulaR1C1 = “=RC[-2]+RC[-1]”
13      Range(“D1”).Select
14      ActiveCell.FormulaR1C1 = “Hallo Welt”
15      Range(“D1”).Select
16      Selection.Font.ColorIndex = 3
17 End Sub
```

Zur besseren Erklärung, wurden die Zeilennummern eingefügt. Im wirklichen Quellcode haben Visual Basic Programme **keine Zeilennummern** mehr. Ältere Basic-Dialekte (beispielsweise GW-Basic) benötigten Zeilennummern zur Programmflusssteuerung.

Die Befehle in den Zeilen 02 bis 05 und in Zeile 07, beginnend mit einem Hochkomma, sind **Kommentarzeile**. Alles was hinter einem Hochkomma steht wird vom Programm ignoriert, auch wenn in diesen Zeilen ansonsten gültige Befehle stehen würden.

Kommentare sind eine sehr wichtige Sache. Sie benötigen zwar einigen Speicherplatz, jedoch sind die Erklärungen nicht nur für andere Programmierer von Interesse, sondern auch für den Verfasser selbst. Wer ein eigenes Programm einmal nach längerer Zeit zur Überarbeitung wieder in die Finger bekommt, ist für sich selbst dankbar, wenn vor jedem Abschnitt steht, was dieser Programmteil eigentlich tut. Anstelle des Hochkommas kann auch der Befehl **REM** verwendet werden – aus dem Englischen kommend für Remark (Bemerkung).

Der Befehl in Zeile 01 „Sub Makro1()“ besagt, dass hier ein Programmabschnitt beginnt, der den Namen Makro1 besitzt. Für jedes einleitende SUB muss am Ende des Abschnitts ein END SUB folgen (hier in Zeile 17). Durch die Marke „END SUB“ wird festgelegt, dass der Programmablauf des Abschnitts hier zu Ende ist. So lassen sich in einem Segment des Visual Basic Editors eine Vielzahl von Programmabschnitten abbilden, ohne dass sich die Programmteile ins „Gehege“ kommen.

Der Befehl in Zeile 08 „ActiveCell.FormulaR1C1 = „120““ besagt, dass die derzeit aktive Zelle mit dem Wert 120 versehen wird. Wir sind im Beispiel davon ausgegangen, dass sich die Schreibmarke genau in der Zelle A1 befindet. Wie hier zu erkennen ist, hat der Programmverlauf dieser Annahme nicht Rechnung getragen. Verändern Sie einmal die Position der Schreibmarke in der Tabelle (Sie können zwischen dem Visual Basic Editor und der Microsoft Excels Tabellenansicht einfach mit der
Taskleiste hin- und herwechseln). Nachdem die Position verändert wurde, sollte das Makro erneut ausgeführt werden und – tatsächlich – es wird immer die aktuelle Position der Schreibmarke mit dem Wert 120 versehen.

In Zeile 09 findet sich der Befehl „Range(„B1“).Select“. Was soviel bedeutet wie, die Zelle B1 wird anvisiert und als aktive Zelle markiert. Ein künftiger Befehl, ohne genaue Positionsanweisung wird sich auf diese Zelle auswirken. Ein solcher Befehl folgt auch: ActiveCell.FormulaR1C1 = „130“. Wie der erste Befehl aus Zeile 08 bewirkt diese Anweisung die Zuweisung eines Wertes, hier 130.

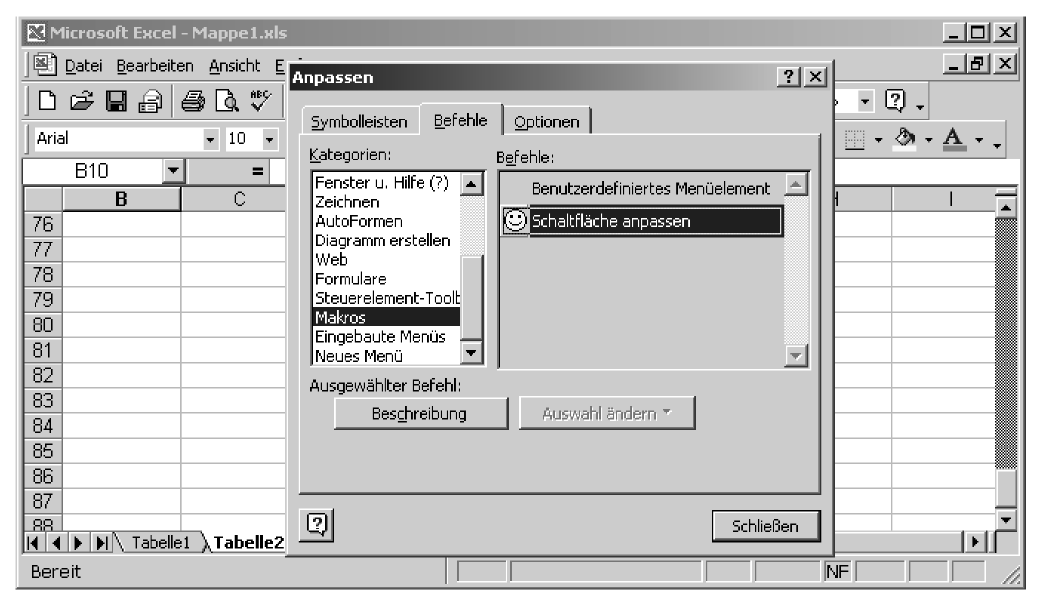
Es folgt das Anvisieren der Zelle C1 und das Einschreiben der Funktion ActiveCell.FormulaR1C1 = „=RC[-2]+RC[-1]“. Gelesen bedeutet der Befehl bewege dich von der aktuellen Zelle zwei Spalten (C = Column) nach links und addiere den Wert aus der Zelle, die von der jetzigen eine Spalte versetzt nach links liegt. Die Befehle in den Reihen 13 bis 15 sind bereits bekannt. Lediglich die Zuweisung einer Farbe in der Zeile 16 ist noch unbekannt:

```
Selection.Font.ColorIndex = 3
```

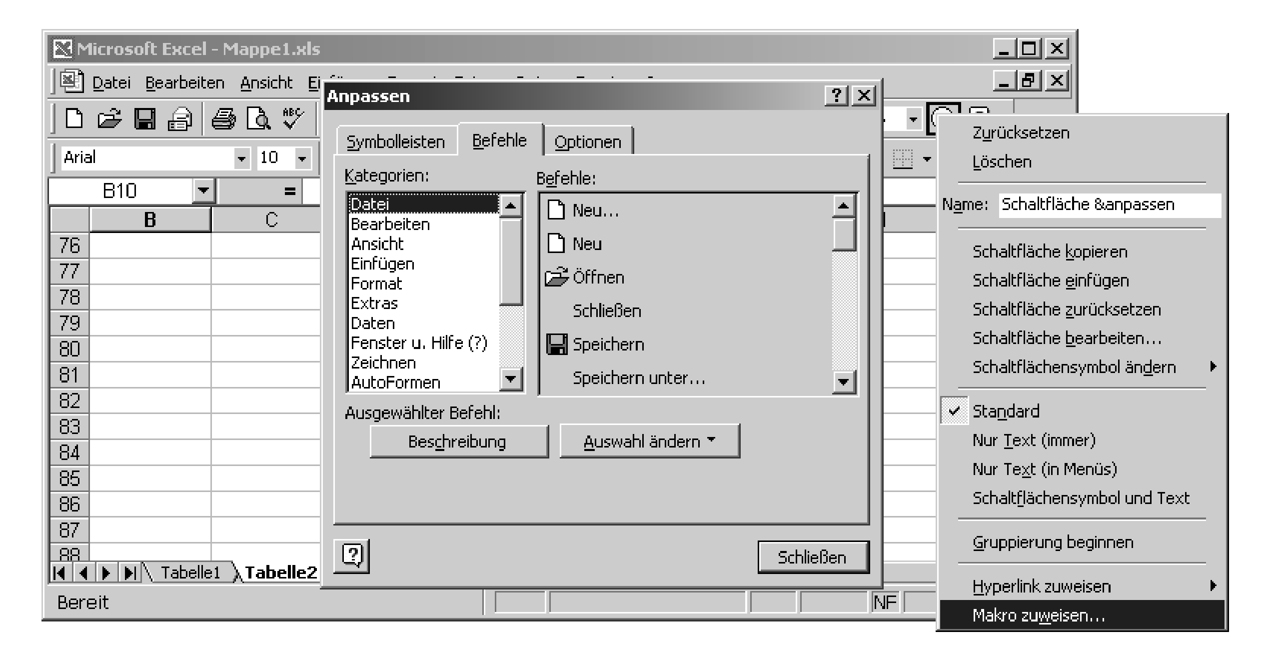
Sprich die aktuelle Selektion (die Auswahl, die in Zeile 15 getroffen wurde besagt die Zelle D1), dort eine Einstellung aus dem Bereich Schriften (FONT) und aus dem Bereich der Schriften die Farbeinstellung ist auf den Wert 3 zu setzen. Das war also ein erstes Makro und die dazugehörige Erklärung. Wie bereits eingangs erwähnt, gibt es viele Methoden im Quellcode ein Ziel zu erreichen. Das wirkliche praktische am Makrorekorder ist, dass er fast ein Nachschlagewerk für Befehle und Eigenschaften ersetzt. Wenn einem ein Eigenschaft mal nicht bekannt ist öffnet man sich einfach eine weitere Excel-Ansicht und zeichnet ein Makro mit den gewünschten Befehlen auf.

11.2 Dem Makro ein Symbol zuordnen

Die Auswahl des Makros ist über das Menü „Extras“ wenig komfortabel. Makros lassen sich wie alle anderen Befehle problemlos in die Symbolleiste einbinden. Mit einem Rechtsklick in die Symbolleiste und anschließendem Klick auf „Anpassen“ wird die Bearbeitung der Symbolleisten eingeleitet.



Nachdem das Fenster „Anpassen“ angezeigt wird, ist das Register „Befehle“ zu wählen und in der Kategorie „Makros“. Mit gedrückter linker Maustaste lässt sich der Smiley in die Symbolleiste ziehen.



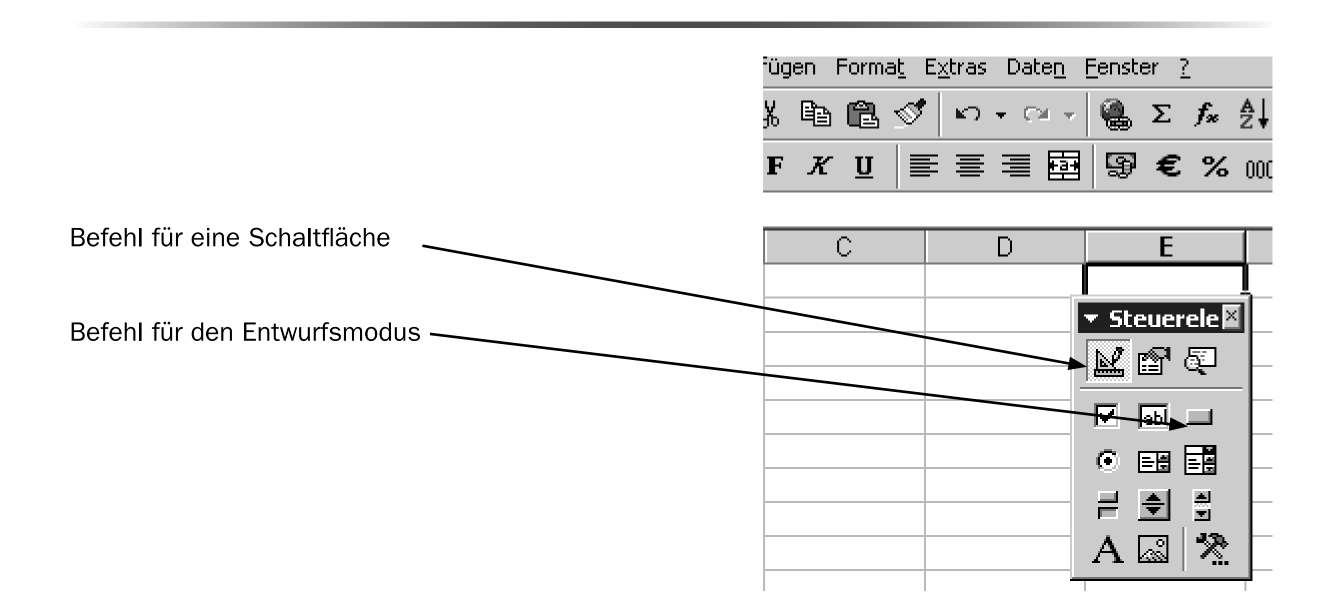
Durch einen Rechtsklick auf das neue Smiley-Symbol wird das Menü für das Symbol angezeigt. Wie man der obigen Abbildung entnehmen kann, lässt sich das Aussehen eines Symbols mit Hilfe eines kleinen Grafikeditors bearbeiten oder anstelle eines Ikons ein Text anzeigen. Der letzte Befehle in der Auflistung „Makro zuweisen...“ bietet die Möglichkeit dem Bildchen das soeben erstellte Makro zu hinterlegen. Anschließend ist ein Klick auf das Symbol ausreichend, um den Quelltext des Makros ablaufen zu lassen.

11.3 Schaltflächen

Neben den Symbolbefehlen bieten sich noch Schalflächen (Buttons) für die Entgegennahme von Benutzerbefehlen an. Schaltflächen sind im Vergleich zu den Symbolen ein wenig unflexibel – Symbole lassen sich beliebig gruppieren oder nach eigenen Vorstellungen anpassen. Diese Flexibilitäten auf der einen Seite ist gleichzeitig das Argument für den Einsatz von Schalflächen. Dem im Gegensatz zu den Symbolen bleiben die Schaltlfächen immer dort, wo sie der Entwickler abgelegt hat. Wer die Befehle kennt, um Schalflächen zu bewegen, sollte auch genügend Kenntnisse haben, diese wieder „richtig zu platzieren“. Symbolbefehle werden von Anwendern mit nur geringen Kenntissen versehentlich ausgeblendet und schon sind sie nicht mehr in der Lage mit der Tabelle zu arbeiten. Im DPE3 wurde bewusst auf den Einsatz von Schaltflächen gesetzt, um diese Probleme erst gar nicht entstehen zu lassen.

In diesem kurzen Abschnitt wird die Einrichtung einer Schaltflächen demonstriert und anschließend der Schaltfläche ein Makro zugewiesen. Ob dieses Makro nun mit dem Makrorekorder, oder mit dem Editor Freihand erstellt wurde, spielt dabei keine Rolle.

Um eine Schaltfläche, oder eine andere typische Windows-Befehlsstruktur, wie beispielsweise Checkboxen oder Radio-Buttons, in eine Tabelle einzublenden, wird das Menü „Steuerelemente Toolbox“ eingeblendet.



Sobald die Toolbox eingeblendet ist, springt Microsoft Excel automatisch in den Entwurfsmodus, das Symbol mit dem Geo-Dreieck, Lineal und Stift ist gedrückt dargestellt. Sollte der Entwurfsmodus nicht aktiv sein, so klicken Sie einfach das Symbol an.

Der Entwurfsmodus wird dazu verwendet aktive Excel-Tabelle zu bearbeiten. Würde der **Entwurfsmodus** nicht aktiviert sein, so könnte eine Schaltfläche beispielsweise gar nicht angeklickt (sprich ausgewählt) werden, ohne, dass der dort hinterlegte Quellcode aktiviert wird.

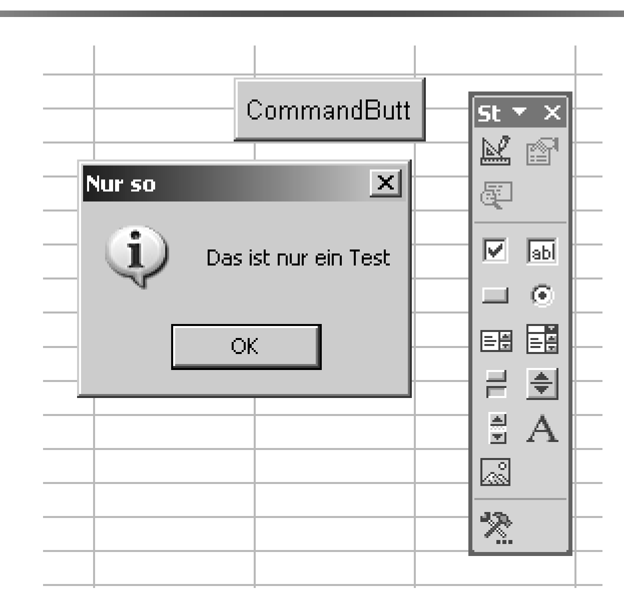
Um nun eine Schaltfläche auf der Tabelle abzulegen ist das Symbol für „Schaltfläche“ anzuklicken und abschließend ein Klick in den Tabellenbereich notwendig. Um die Schaltlfäche zu positionieren, ist der Mauszeiger direkt auf dem Objekt zu positionieren bis der Mauszeiger zum „**Navigationspfeil**“ (Pfeile in die vier Himmelsrichtungen) wird. Wird nun die linke Maustaste gedrückt gehalten und die Maus gleichzeitig bewegt (Befehlsform: ziehen), so lässt sich das Objekt frei positionieren. Die Dimensionen der Schaltlfäche werden über die sechs weißen Kästchen an den Rändern verändern.

Mit einem Doppelklick auf der Schaltlfäche wird die Entwicklungsumgebung von Visual Basic geöffnet, die im Abschnitt 12 näher erläutert wird. Im Programmcode befindet sich die Schreibmarke nun sofort am richtigen Ort, um einen Befehl einzugeben. VBA stellt automatisch einen durchnummerierten Abschnitt im Code, ein sogenantes Unterprogramm (beginnend mit Private Sub) zur Verfügung. Alles zwischen dem Private Sub und dem End Sub wird im Folgenden ausgeführt, sobald die Schaltfläche geklickt wird.

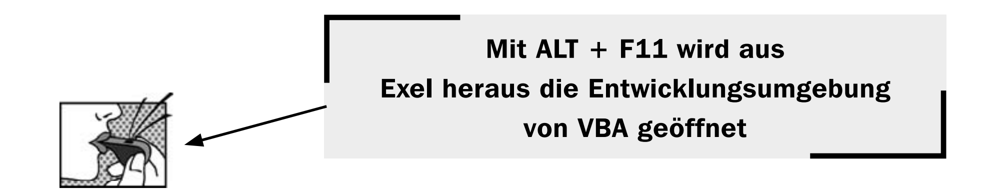
```
msgbox „Das ist nur ein Test“, vbInformation, „Nur so“
```

Wird der oben gezeigte Befehl `msgbox` eingegeben, so erscheint während des Schreibens die Eingabehilfe IntelliSense von Microsoft, die Auswahlmöglichkeiten zum Befehl anbietet.

Wird die Entwicklungsumbegung von Microsoft Visual Basic nach der Eingabe des Befehls geschlossen und der Entwurfsmodus mit einem Klick auf „Geo Dreieck“ beendet steht der Befehl zur Laufzeit zur Verfügung. „Zur Laufzeit“ ist die Bezeichnung für den normalen Betrieb eines Programms.

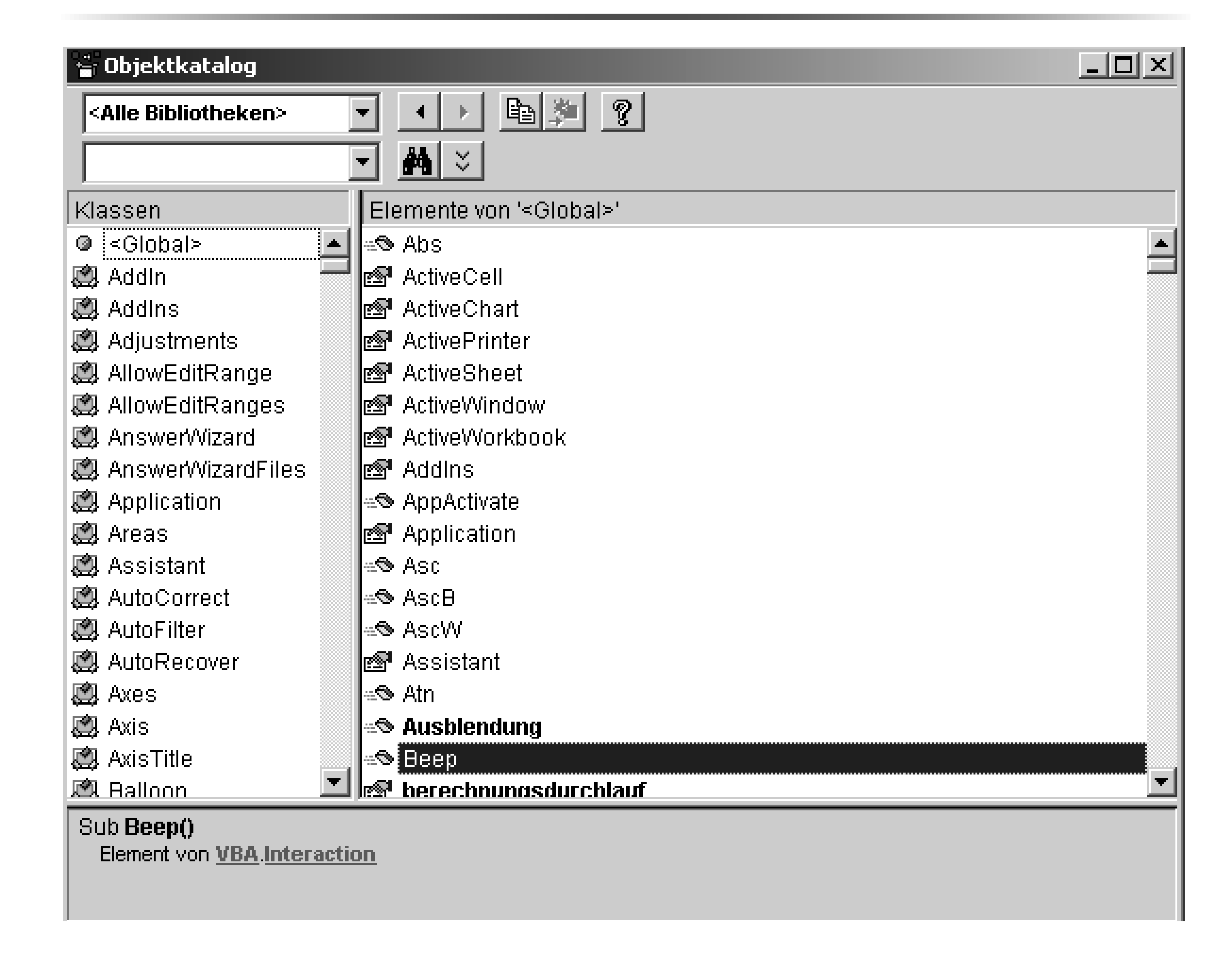


Bis der Benutzer auf „OK“ klickt ist der Programmablauf im VBA-Code unterbrochen. Weitere Informationen zur Programmierung finden sich im folgenden Abschnitt. Um Einstellungen und Dimensionen der Schaltfläche zu verändern ist der Entwurfsmodus wieder zu aktivieren. Mit einem Rechtsklick auf die Schaltfläche lässt sich beispielsweise festlegen ob diese bei einem Ausdruck mit auf´s Papier gebracht wird. Um die Beschriftung der Schaltlfäche zu verändern ist ein einfacher Klick in den Titel notwendig.



11.4 Entwicklungsumgebung

Das folgende Kapitel betrachtet Funktionen in VBA die im „Dienstplan mit Excel“ verwendet wurde, um eine elektronisch gestützte Dienstplanung zu ermöglichen. Exemplarisch werden diese Funktionen erklärt. Weiterführende Informationen finden sich in den im Anhang genannten Programmier-Büchern für Microsoft Excel. Eine komplette Übersicht aller Befehle von VBA würde hier bei weitem den Rahmen sprengen. Zudem finden sich die Befehle und ihre Optionen auch in dem **Objektkatalog** in der Entwicklungsumgebung, der sich mit einem Druck auf die Taste F2 öffnen lässt.



In der Online-Hilfe von Microsoft Visual Basic, die mit einem Druck auf die Taste F1 eingeblendet werden kann, findet sich im Inhalt unter „Hilfe zu Visual Basic Benutzeroberfläche“ eine umfassende Einführung zu diesem Bereich. Da die Online-Hilfe in jeder Office-Installation zur Verfügung steht, soll hier nicht im Detail auf die Bedienung und die Befehle eingegangen werden.

Die Mehrzahl von Beispielen basieren auf einer oder zwei Schaltflächen auf einer Excel-Tabelle. Wie eine Schaltfläche erzeugt und diese mit einem leeren Quellcode-Gerüst versehen wird, ist in Abschnitt 11.3 erklärt.

Egal wie groß der verwendete Monitor auch ist, er ist stets zu klein. Um zumindest die häufige Nutzung der Rollbalken zu verhindern, besteht die Möglichkeit mitten in einer laufenden Befehlszeile einen **Umbruch** einzufügen. So sind beispielsweise beide Quelltexte von gleicher Funktion:

```
Private Sub CommandButton1_Click()
MsgBox „Hallo Welt, ich wollte nur sagen mir gefällt`s“
End Sub
```

oder

```
Private Sub CommandButton1_Click()
MsgBox “Hallo Welt, ich wollte nur sagen mir gefällt`s” _
vbInformation, “Titel des Dialogfensters”
End Sub
```

Der **Unterstrich** hiner dem Anführungszeichen sorgt für einen Umburch innerhalb der gleichen Programmzeile. Beide Zeilen werden wie eine einzige Zeile behandelt.

11.5 Variablen

Variablen werden die meisten noch aus der Schulzeit kennen: a + b = c. Variablen sind Stellvertreter für Werte. Doch im Gegensatz zum Mathematik- und Physikunterricht wird in der Programmierung nicht nur ein Datentyp für Zahlen vorgehalten, sondern für alle möglichen Datenformen (natürlich werden alle Daten intern vom Computer wieder als Zahlenwert aus Bits und Bytes verwaltet).

Üblicherweise erfordert Microsoft Excel in VBA keine **Deklaration** von Variablen, sondern erlaubt eine beliebige Zuordnung in einem speziellen Datentyp, der als VARIANT (quasi variabel) bezeichnet wird. Der Vorteil einer manuellen Deklaration durch den Programmierer ist die Sicherstellung, dass im Quelltext nicht versehentlich der Variablenname verwechselt wurde. Aus diesem Grunde beginnen alle ordentlich programmierten Quelltexte unter Visual Basic mit der Anweisung `OPTION EXPLICIT`. Nachdem diese Option ausgesprochen wurde, ist eine Deklaration einer Variable mit dem Befehl DIM erforderlich.

Dabei stellt sich natürlich die Frage, was der Vorteil einer Deklaration nach einem Dateityp gegenüber dem automatischen VARIANT ist? Der Datentyp VARIANT verbraucht im Vergleich zu allen anderen Datentypen am **meisten Speicherplatz** und wird somit in der Verarbeitungsgeschwindigkeit am langsamsten abgearbeitet. Das macht bei einem Zugriff auf wenige Informationen auf aktuellen Rechnern sicherlich nicht soviel aus – aber wenn es der Programmierer verhindern kann, sollte er von Haus dafür sorgen, dass eine Ablaufgeschwindigkeit so optimal wie nur möglich ist. Bei der Benennung von Variablen sind einige Regeln zu beachten: Keine Leerzeichen, Punkte oder Sonderzeichen. Deutsche Umlaute sind erlaubt, jedoch wird zwischen Groß- und Kleinschreibung nicht unterschieden. Zudem dürfen Variablen-Namen nicht mit VBA-Schlüsselworten, beispielsweise Sub, End, For oder Dim, übereinstimmen.

```
DIM A AS STRING
DIM B AS INTEGER
DIM C,D,E AS LONG
```

Achtung! Die dritte Dimensionierung im obigem Beispiel hat nur C den Datentyp LONG zugeordnet, während D und E als VARIANT deklariert wurden!

Datentyp	Bereich	Speicherbedarf
Byte	0 bis 255	1 Byte
Boolean	True (Wahr), False (Unwahr)	2 Byte
Integer	-32768 bis +32767	2 Byte
Long	-2147484648 bis +2147483647	4 Byte
Currency	Festkomma 15 Stellen, 4 Stellen	8 Byte
Single	Fließkommazahl mit 8 Stellen Genauigkeit	4 Byte
Double	Fließkommazahl mit 16 Stellen Genauigkeit	8 Byte
Date	Datum, Uhrzeit	8 Byte
String	Zeichenkette max. 2 GByte	10 Byte + 2 je Zeichen
Variant	Beliebig	min. 16 Byte, bei Zeichenketten 22 + 2 Byte

Aus älteren Basic-Dialekten stammt die alternative Deklaration einer Variablen mit einem Sonderzeichen:

%	Integer
&	Long
@	Currency
#	Double
!	Single
$	String

Ob eine Variable lediglich innerhalb des Unterprogramms, oder im ganzen Projekt gültig ist, bestimmt der Ort und die Art der Deklaration.

```
Public a As String

Private Sub CommandButton1_Click()
Dim b As String
a = „Hallo A“
b = „Hallo B“
MsgBox a & „:“ & b & „:“ & c
End Sub

Private Sub CommandButton2_Click()
Dim c As String
c = “Hallo C”
MsgBox a & “:” & b & “:” & c
End Sub
```

Werden, wie in diesem Beispiel beschrieben, zwei Schaltflächen abgesetzt und zunächst die Schaltfläche 2 gewählt, so erscheint als Ausgabe ::Hallo C. Wird anschließend die Schaltfläche 1 gedrückt, so erscheint Hallo A:Hallo B:. Durch die Deklaration der Variable a mit PUBLIC gilt diese innerhalb des gesamten Projekts. Da jedoch zunächst Schaltfläche 2 gedrückt wurde, war a noch kein Wert zugewiesen, aus diesem Grunde wurde mit der MSGBOX nur der Wert aus c ausgegeben.

Die Deklaration der Variablen haben Auswirkungen auf die „Verbindbarkeit“ von Daten. VBA ist jedoch sehr flexibel und beantwortet selbst folgenden Programmcode korrekt:

```
Option Explicit

Private Sub CommandButton1_Click()
Dim a As Integer
Dim b As String
Dim c As String
a = 400
b = „400“
c = b + a
MsgBox c
End Sub
```

Wird die Zeile c = b + a durch c = b & a ersetzt, so erscheint als Ergebnis 400400, da beide Werte als Zeichenkette miteinander verbunden werden. Es sei nur am Rande erwähnt: Andere Programmiersprachen machen solche „Deklarationseskapaden“ nicht mit und erzeugen brav einen Laufzeitfehler. Visual Basic erlaubt auch die Deklaration komplett eigener Datenfelder, da diese im DPE3 jedoch nicht verwendet werden, soll eine Erklärung an dieser Stelle entfallen.

Felder

Felder sind so etwas wie Listen von Variablen gleichen Namens, die über Indexnummern angesprochen werden. Felder werden dann eingesetzt, wenn mehre ähnliche Informationen in einer Matrix abgelegt werden sollen. Wie Variablen, so müssen auch Felder (im Englischen Arrays) deklariert werden.

```
Private Sub CommandButton1_Click()
Dim DIENSTFORM(6) As String
Dim Ausgabe As String
Dim A As Integer
For A = 1 To 6
DIENSTFORM(A) = Cells(3 + A, 6).Value
Ausgabe = Ausgabe & DIENSTFORM(A)
Next A
MsgBox Ausgabe
Msgbox DIENSTFORM(3)
End Sub
```


Finden sich im Bereich der aktuellen Tabelle zwischen F4 und F9 Bezeichnungen für Dienstformen, so lassen sich diese in das Array einlesen. Der erste MSGBOX-Befehl gibt den zusammengesetzten Block von Dienstformen aus und der anschließende MSGBOX-Befehl nur die dritte Dienstform.

11.6 Operatoren

Die meisten Operatoren sind bereits benannt und genutzt worden, ohne das man sich besondere Gedanken um sie gemacht hat. Opetrativ werden Addition (+), Subtraktion (-), Division (/) und Multiplikation (*) sehr oft beim Umgang mit Microsoft Excel verwendet. In die Gruppe der Operatoren gehören jedoch ferner das Prozentzeichen (%) und für die Potenierung das Caret (^).

Weiter bietet Microsoft Excel verschiedene Verlgeichsoperatoren :

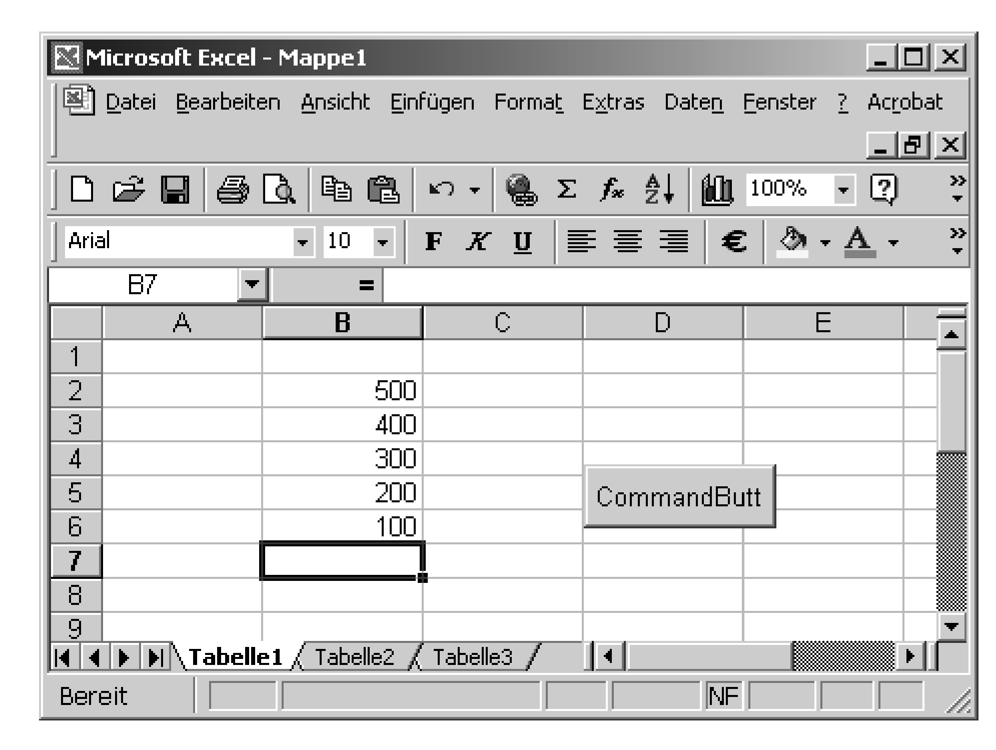
=	Gleich	A1 = B1
>	Größer als	A1 > B1
<	Kleiner als	A1 < B1
>=	Größer oder Gleich	A1>=B1
<=	Kleiner oder Gleich	A1<=B1
<>	Ungleich	<>

Werden Textfragmente bearbeitet, so wird für die Verbindung nicht das Pluszeichen, sondern das kaufmännische Und-Zeichen (&) verwendet.

```
„Thomas“ & „Bär“ ergibt ThomasBär
```

11.7 Zugriff auf Zellen

Für eine sinnvolle Zusammenarbeit zwischen VBA-Code und der Excel-Tabelle ist es natürlich notwendig, dass vom Programmcode aus auf die Werte in den Tabellen zugegriffen werden kann.




```
Private Sub CommandButton1_Click()
MsgBox Worksheets(“Tabelle1”).Range(“B2”).Value
End Sub
```

Mit Worksheets(„Tabelle1“).Range(„B2“).Value wird auf den Wert in der Zelle B2 in der Tabelle 1 verwiesen. Diese Vorgehensweise ist vor allem dann interessant, wenn auf eine Tabelle zugegriffen wird, die nicht notwendigerweise aktuell im Vordergrund geöffnet ist. Ist im Programmverlauf geregelt, dass nur die aktuelle Tabelle gemeint sein kann, so lässt sich der Befehl verkürzen.

```
Private Sub CommandButton1_Click()
MsgBox Cells(2, 2).Value
End Sub
```

Im Gegensatz zu RANGE lässt sich CELLS auch für Schleifenbefehle nutzen, dazu mehr im folgenden Abschnitt.

Über die gleiche Vorgehensweise lassen sich auch Werte und Eigenschaften von Zellen verändern. Beispielsweise soll der Schrifttyp für den ganzen Bereich auf Fett umgestellt werden.

```
Private Sub CommandButton1_Click()
Range(Cells(2, 2), Cells(7, 2)).Font.Bold = True
End Sub
```

11.8 Schleifen

Schleifen sind etwas ganzes Besonderes in der Programmierung. Mit ihnen lassen sich wiederkehrende Schritte bis zur Erfüllung einer Bedingung wiederholen. VBA kennt mehrere Schleifenarten, die bekannteste Schleifenform ist die FOR NEXT Schleife.

In Anlehnung an das Beispiel aus dem vorherigen Abschnitt, soll eine solche Schleife nun dafür genutzt werden, die Werte zu addieren.

```
Private Sub CommandButton1_Click()
Dim a As Integer
Dim e As Integer
For a = 1 To 6
e = e + Cells(1 + a, 2).Value
Next a
MsgBox e
End Sub
```

Zunächst werden die Variablen a und e als ganzzahlige Werte dimensioniert. Da keine Kommawerte in den Zellen vorkommen, ist dies der optimale Datentyp. Die Schleife beginnt mit dem Befehl FOR A = 1 TO 6. Übersetzt heißt dies, Zähle den Wert in der Variable A von 1 bis 6. Die Befehle zwischen FOR und NEXT werden nun sechsmal wiederholt. Da im Beispiel die Wertauflistung in B2 beginnt, wird im Befehl CELLS mit einem 1 +a dafür gesorgt, dass die Zählung der Werte bei 2 beginnt und nicht bei 1, wie in der Schleife festgelegt.

Würde der Befehl MSGBOX e vor das NEXT A verschoben werden, so müsste der Benutzer zur Laufzeit, nach dem Klick auf die Schaltfläche sechsmal das Dialogfeld mit dem stetig steigenden Wert e mit OK bestätigen.

Vorsicht : Im Gegensatz zu den Befehlen in Microsoft Excel selbst, wird in VBA keine Überprüfung von logischen Programmfehlern vorgenommen. Würde vor dem NEXT A ein Befehl stehen wie beispielsweise A = 1, so würde die Bedingung A=6, sprich das Ende der Schleife niemals erreicht, und der Computer würden „hängen bleiben“.

Mit dem Befehl STEP n lässt sich festlegen, in welchen Schritten die FOR NEXT Schleife fortgezählt werden soll – dies ist üblicherweise der Wert 1. Möchte man erreichen, dass nur jede zweite Zeile gezählt wird, so kann die Schleife entsprechend umgestaltet werden:

```
For a = 1 To 6 step 2
```


Es sind auch Schritte kleiner 1 zulässig – nicht jedoch im hier beschriebenen Beispiel, da eingangs für die Variable a der Datentyp INTEGER, sprich Ganzzahl, festgelegt wurde.

Weitere Schleifenformen wir „DO LOOP“ oder „WHILE WEND“ werden in der Literatur und der Online-Hilfe von VBA (Taste F1) umfassend erläutert und sollen hier, da im DPE3 nicht verwendet, nicht benannt werden. Beide Schleifentypen eignen sich hervorragend für Bedingungen, deren berechnender Verlauf nicht in sich wiederholenden Schritten, sondern in der Erfüllung der Bedingung an sich begründet sind. Sprich die Anzahl notwendiger Schritte zu Beginn der Berechnung noch unklar ist.

11.9 Strukturen

Nicht immer kann ein Programm von vorn bis hinten durchlaufen werden, ohne das einige Programmschritte übersprungen werden. Die Befehle CALL, GOTO und GOSUB ermöglichen den Programmverlauf zu verändern. Ähnliches gilt für selbstdefinierte Funktionen, die aber im DPE3 nicht verwendet wurden und aus diesem Grunde hier nur genannt werden sollen.

Ein besonders wichtige, strukturgebende Befehlsfolge ist IF THEN ELSE !

Das Beispiel aus dem vorherigen Abschnitt soll verändert werden und zwar dahingehend, das immer, wenn die Schleife einen ungeraden Wert hat a=1, a=3 und a=5, dann soll der ausgelesene Wert aus der Tabelle addiert und wenn die Schleife einen ungeraden Wert hat, so soll subtrahiert werden.

```
Private Sub CommandButton1_Click()
Dim a As Integer
Dim e As Integer
For a = 1 To 6
If a = 1 Or a = 3 Or a = 5 Then
e = e + Cells(1 + a, 2).Value
Else
e = e - Cells(1 + a, 2).Value
End If
Next a
MsgBox e
End Sub
```

Wird die Bedingung nach dem IF THEN Befehl erfüllt, so werden die Befehle vor dem ELSE ausgeführt, ansonsten die Befehle nach dem ELSE. Die IF THEN ELSE Bedingung muss mit einem END IF abgeschlossen werden. Lässt sich der IF-Befehl problemlos in einer einzigen Zeile abbilden, so kann das END IF entfallen – sofern kein ELSE benötigt wird. Beispielsweise soll hinter dem END IF eine zweite IF-Abfrage den Benutzer darüber informieren, dass der Wert VIERHUNDERT gefunden wurde :

```
If e = 400 Then MsgBox „Ist Vierhundert“, vbInformation
```

In diesem Fall wird alles in eine einzige Zeile geschrieben.

Sprungbefehle

Der GOTO Befehl steht immer wieder in der Kritik der „ordentlichen Programmierer“, da ein einfacher Sprungbefehl ein Zeichen für einen nicht sehr optimierten Code darstellt. Für das Ziel, einen Dienstplaner mit Microsoft Excel zu entwerfen, lässt der eine oder anderer GOTO-Befehl schon einmal einsetzten, ohne das jemand mit der gelben Karte winkt.

```
Private Sub CommandButton1_Click()
Dim a As Integer
a = 1
MsgBox a
GoTo weiter
a = 2
MsgBox a
```

```
weiter:
a = 3
MsgBox a
End Sub
```

Bei obigen Beispiel wird der Wert 2 niemals ausgeben, da mit dem Befehl GOTO WEITER einfach über die zwei Programmschritte vor dem Befehl WEITER: hinweggesprungen wird.

Ein enger Verwandter der GOTO-Befehls ist der GOSUB-Befehl, der den Programmablauf in einem Unterprogramm fortführt, dann jedoch zurück in das Hauptprogramm springt.

```
Private Sub CommandButton1_Click()
Dim a As Integer
a = 1
MsgBox a
GoSub Unterprogramm
MsgBox a
a = 2
MsgBox a
Exit Sub
Unterprogramm:
a = 3
Return
End Sub
```

Obiges Beispiel sorgt für die Ausgabe von 1, dann 3 und abschließend 2. Mit dem Befehl GOSUB springt das Programm in den Bereich hinter Unterprogramm: und wird nach dem Befehl RETURN mit dem MSGBOX nach dem GOSUB fortgeführt. Die Verwendung des Gosubs gilt gemeinhin auch unter Aufsicht gestandener Programmierer als nicht verpönt.

```
Public a As Integer

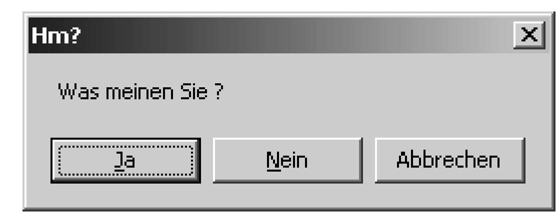
Private Sub CommandButton1_Click()
a = 1
MsgBox a
Call weiter
MsgBox a
a = 2
MsgBox a
End Sub

Sub weiter()
a = 3
End Sub
```

Mit dem Befehl CALL wird auch in ein Unterprogramm gesprungen, dieses Unterprogramm steht jedoch dem gesamtem Projekt zur Verfügung. Nach dem CALL springt der Ablauf in das SUB WEITER, führt die Befehle aus und springt wieder an die Stelle zurück, von der der CALL aufgerufen wurde. Der Befehl CALL bietet viele interessante Möglichkeiten, da ihm auch direkte Werte mitgegeben werden können. An dieser Stelle sei ausdrücklich auf die weiterführende Literatur verwiesen!

11.10 Eingabemöglichkeiten

Oft muss der Benutzer befragt werden, wie ein Programmverlauf fortgesetzt werden soll, oder was für ein Wert verwendet werden soll. VBA bietet die üblichen Dialogfelder, die aus Microsoft Windows bekannt sind – interessanterweise jedes Mal in der spezifischen Art und Weise der jeweiligen Windows-Version.

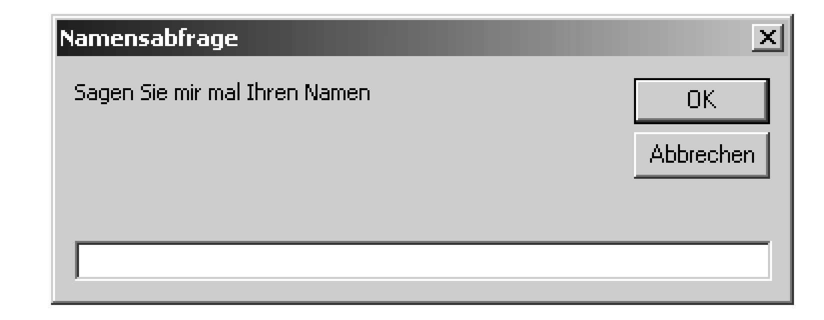



```
Private Sub CommandButton1_Click()
a = MsgBox(„Was meinen Sie ?“, vbYesNoCancel, „Hm?“)
If a = vbYes Then MsgBox „Ja wurde gedrückt.“
If a = vbNo Then MsgBox „Nein wurde gedrückt.“
If a = vbCancel Then MsgBox „Es wurde Abbrechen gedrückt.“
End Sub
```

Für die Antworten werden konstante Variablennamen in Visual Basic verwendet. In einigen Bereichen des DPE3-Quelltexts wird die Variable a als STRING umgesetzt, wobei NEIN dem Wert 7 entspricht.

Welche Antworten im Dialogfeld angeboten werden, wird über eine weitere Konstante von VB gesteuert. vbAbortRetryIgnore erzeugt beispielsweise ein Dialogfeld mit der Möglichkeit „Abbrechen“, „Wiederholen“, „Ignorieren“, während vbCritical ein Dialogfeld mit der Möglichkeit „OK“ zu klicken bietet, jedoch mit dem roten Warnhinweis-Symbol. Welche Variationen von „Dialog-Styles“ es gibt, kann über den Objektkatalog (Taste F2) oder über die Eingabehilfe Intelli-Sense herausgefunden werden.

Soll der Benutzer einen Wert angeben, so empfiehlt sich eine andere Dialogform:




```
Private Sub CommandButton1_Click()
a = InputBox(„Sagen Sie mir mal Ihren Namen“, „Namensabfrage“)
If a = „“ Then Exit Sub
MsgBox „Hallo „ & a & „, ich grüße Sie herzlichst.“
End Sub
```

Es lassen sich auch „exotischere“ Eingabefenster mit Hilfe des Dialogeditors aufbauen. Diese Vorgehensweise ist jedoch eher dann erforderlich, wenn die Standarddialoge nicht ausreichen sollten.

11.11 Fehler abfangen

VBA bietet einen Fehlerhandler, mit dem auftretende Fehler abgefangen werden können. Da üblicherweise ein auftretender Fehler den Debugger, sprich die Entwicklungsumgebung mit der Möglichkeit Variablenwerte auszulesen, in den Vordergrund bringt, ist dies in vielen Fällen unerwünscht. Es macht für den späteren Anwender eines Dienstplans auch einen besseren Eindruck, wenn in einem Fehlerdialog ein für die Einrichtung passender Dialog erscheint. Beispielsweise „Tritt dieser Fehler öfters auf, setzten Sie sich bitte mit Herrn Mayer (Tel -123) in Verbindung (Fehler: Aufruf3)“.

Ein Fehlerhandler ist schnell eingerichtet:

```
Private Sub CommandButton1_Click()
Dim a As Integer
Dim b As Integer
On Error GoTo fehler
b = 1 / a
```

```
Exit Sub
fehler:
MsgBox "Es ist ein Fehler aufgetreten.", vbInformation, "Uups"
End Sub
```

Da a im obigen Beispiel noch keinen Wert hat, wird 1 durch 0 geteilt, was zu einem Fehler führen muss. Doch anstelle der kryptischen Fehlermeldung von Microsoft Excel folgt die obige Fehlermeldung. Der Programmablauf wird hier mit END SUB ordentlich beendet. Es lässt sich aber auch ein weiterarbeiten des Programms erreichen, wobei nur der fehlerhafte Befehl übersprungen wird:

```
Private Sub CommandButton1_Click()
Dim a As Integer
Dim b As Integer
On Error GoTo fehler
b = 1 / a
b = a / 1
MsgBox b
Exit Sub
fehler:
MsgBox "Fehler: Wir machen einfach mal weiter", vbInformation
Resume Next
End Sub
```

Mit dem Befehl RESUME NEXT wird der Programmablauf fortgesetzt. Diese Möglichkeit sollte jedoch eher als „Notanker" verwendet werden, denn als ordentliche Programmfortführung.

11.12 Dateien schreiben und externe Programme

Es kann notwendig sein, aus einem Programmcode heraus ein anderes Programm zu starten. In den „ausgewachsenen" Programmiersprachen gibt es da die interessantesten Möglichkeiten, um beispielsweise für einen **Informationsaustausch** zwischen zwei Programmprozessen zu sorgen. Im Internet (beispielsweise auf der deutschsprachigen Homepage für das Excel-Forum (www.igelnet.de)) finden sich auch für Visual Basic und VBA sehr spannende „Einsatzmöglichkeiten".

Der Zugriff erfolgt unter VBA am einfachsten mit dem Befehl SHELL, sprich mit einem Zugriff über das Betriebssystem – so, als ob der Befehl direkt auf Betriebssystemsebene gegeben worden wäre.

```
Private Sub CommandButton1_Click()
Shell „Notepad", vbNormalFocus
End Sub
```

Mit obigem Befehl wird das Programm „Notepad" (der einfache Texteditor aus dem Lieferumfang von Microsoft Windows) gestartet und mit der „üblichen" Ansicht ausgeführt. vbMaximizedFocus würde dafür sorgen, dass das Programm „maximiert" ausgeführt wird, sprich den ganzen Bildschirm ausfüllt.

Im DPE3 wird nur ein einziges Programm extern auferufen, und zwar ein Texteditor für die Ausgabe der qualitativen Plananalyse (CAC). Künftig wird auch diese Funktionalität direkt im DPE3 umgesetzt. Im Falle des CAC wird zunächst eine Textdatei erzeugt und mit Daten befüllt. Diese wird nach dem Vorgang geschlossen und mit Hilfe des **SHELL**-Aufrufs mit dem Texteditor betrachtet.

```
Private Sub CommandButton1_Click()
a = FreeFile()
Open „C:\DATEI.TXT" For Output As a
b = InputBox("Geben Sie einen Wert ein, der in die Datei geschrieben werden soll")
Print #a, b
Close a
b = „"
a = FreeFile()
Open „C:\DATEI.TXT" For Input As a
Input #a, b
```



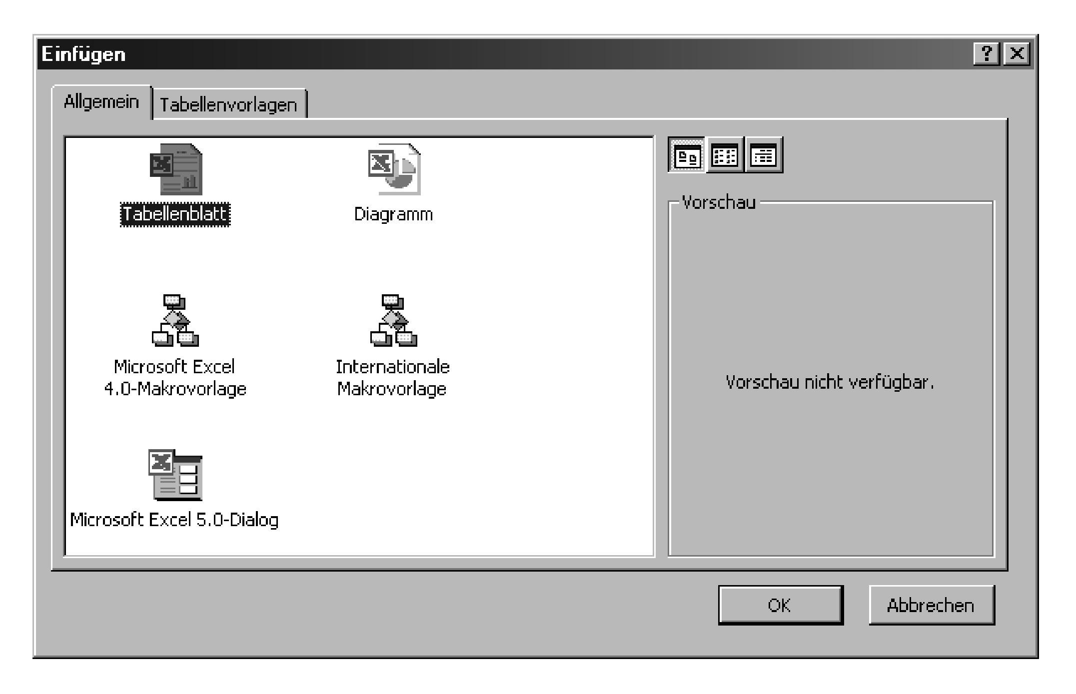
```
Close a
MsgBox b
Shell „Notepad C:\DATEI.TXT"
End Sub
```

Im obigen Beispiel wird eine Datei für das SCHREIBEN geöffnet und mit Dateinummer a versehen. Die Anzahl gleichzeitig geöffneter Dateien wird durch das Betriebssystem und durch die Software (in diesem Fall VBA) begrenzt. Wenn Sie nicht explizit wissen, welche Dateinummer der verwendeten Software frei ist, empfiehlt sich die Vergabe mit FREEFILE – sprich die Zuweisung der nächsten freien Dateinummer durch die Applikation.

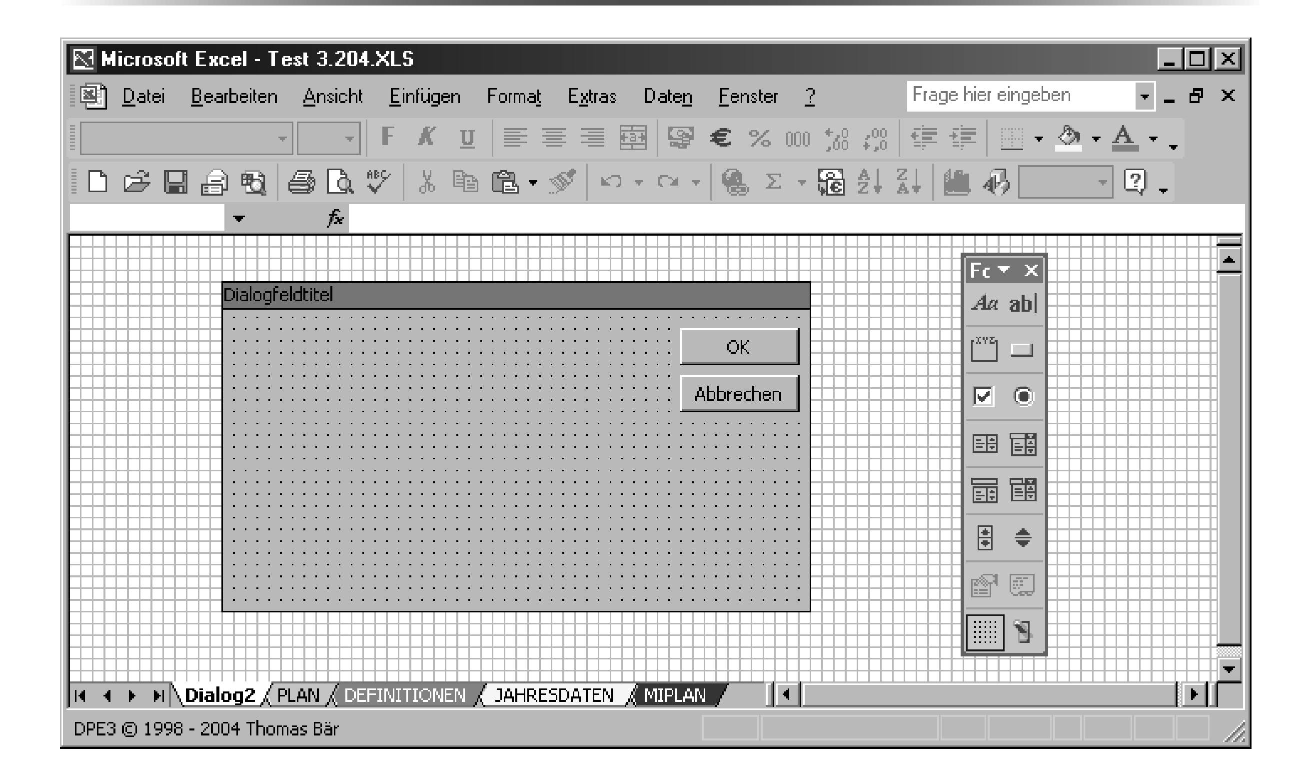
Mit dem Befehl PRINT <DATEINUMMER>,<INFORMATION> wird der Datenstrom in die Datei geschrieben. Mit dem Befehl CLOSE <DATEINUMMER> wird die geöffnete Datei wieder geschlossen. Anschließend wird dem gleichen Vorgang, lediglich für das LESEN geöffnet die gleiche Information wieder dargestellt. Der abschließende Aufruf des Texteditors mit NOTEPAD verdeutlicht, dass die geschriebene Datei nicht in einem spezifischen Format von VBA oder Excel abgelegt wurde, sondern als schlichte ASCII/Text-Datei. So erzeugte Dateien können quasi von jedem Rechnersystem eingelesen werden. Programmdateien und spezifische Datendateien, beispielsweise .XLS-Dateien von Excel, werden nicht im ASCII/Text-Format geschrieben, sondern im so genannten Binärformat.

11.13 Individuelle Dialogfelder

Seit der Version 5 bietet Microsoft Excel die Möglichkeit Dialogfelder direkt im Programm individuell zu Erstellen. In den Dialogfeldern lassen sich alle üblichen Windows-Bedienungselemente einsetzen. Um ein neues Dialogfeld zu Erstellen ist ein Rechtsklick auf die Registerleiste notwendig. Nach einem Klick mit der linken Maustaste auf „Einfügen" erscheint diese Auswahl:



In dieser Auswahl ist der „Microsoft Excel 5.0-Dialog" auszuwählen, und anstelle eines Tabellenblatt in die Excel-Datei aufgenommen.



Die Dimensionen des künftigen Dialogfeld lassen sich, wie bei Fenstern üblich, an den Eckpunkten einstellen. Wird ein sehr großer Monitor verwendet, so muss bedacht werden, dass auf einem kleineren Monitor eventuell Teile des Fensters nicht dargestellt werden.

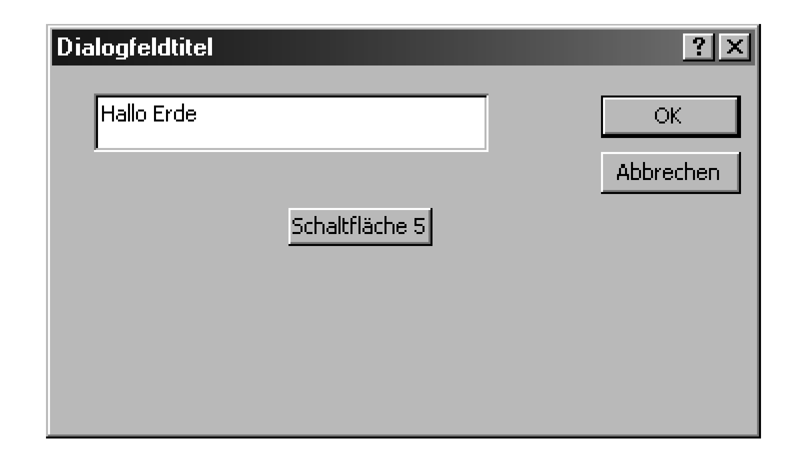
Mit Hilfe der Schaltfläche „ab|“ lässt sich der Mauszeiger in ein Fadenkreuz zur Erstellung eines Bearbeitungsfelds verwandeln. Anfang und Endpunkt müssen mit gedrückter linker Maustaste im Dialogfensterbereich festgelegt werden und Microsoft Excel vergibt automatisch eine Bezeichnung für das Textfeld.

Mit einem Rechtsklick auf den Rahmen des neu erstellen Textfeld lässt sich über „Eigenschaften“ das Eigenschaftmenü für dieses **Textfeld** vorblenden. In diesem Fenster lässt sich festlegen, ob die Eingabe im Feld sichtbar sein soll, oder wie Kennwortfeldern verdeckt geschehen soll. Auch lassen sich Gültigkeitsprüfungen auf Text, Zahl, Bezug oder Formeln einstellen, beziehungsweise ob ein Feld Ein- oder Mehrzeilig sein soll.

Ist das Textfeld markiert und wird der Befehl „Code bearbeiten“ aus der Symbolleiste „Formular“ ausgewählt so springt Microsoft Excel automatisch in den Visual Basic Editor mit dem Eintrag `Sub <NAME DES TEXTFELDES>.Bei_Änderung()`. Dieser Code würde ausgeführt, sobald eine Änderung vorgenommen wird. Um das Fenster auszuprobieren, muss der kleine Schalter unten rechts in der Symbolleiste angeklickt werden.

Wichtig in diesem Zusammenhang ist das Ansprechen des Textfeldes aus dem Quellcode aus anderen Programmbereichen heraus. An dieser Stelle bürdet Visual Basic dem Programmierer ein wenig Tipparbeit auf. Das im obigen Beispiel erzeugte Textfeld ließe sich wie folgt ansprechen:


```
DialogSheets(„Dialog2“).[Bearbeitungsfeld 4].Caption = „Hallo Erde“
```



Die Besonderheit an den Dialogfeldern mit Textfeldern ist, dass sie nicht mit Hilfe .VALUE angesprochen werden, sondern mit .CAPTION.

Eine wichtige Struktur in Dialogfeldern sind die so genannten Radio-Buttons (die deutsche Bezeichnung Optionsfeld klingt deutlich langweiliger) – kleine Knöpfe, wie sie einst am Autoradio zu finden waren: Wird ein Senderknopf gedrückt, springt der zuvor gedrückte Knopf in die Nullstellung zurück. Mit Hilfe des Rahmens wird dafür gesorgt, zu bestimmen, welche Radio-Buttons als eine Einheit zu betrachten sind:

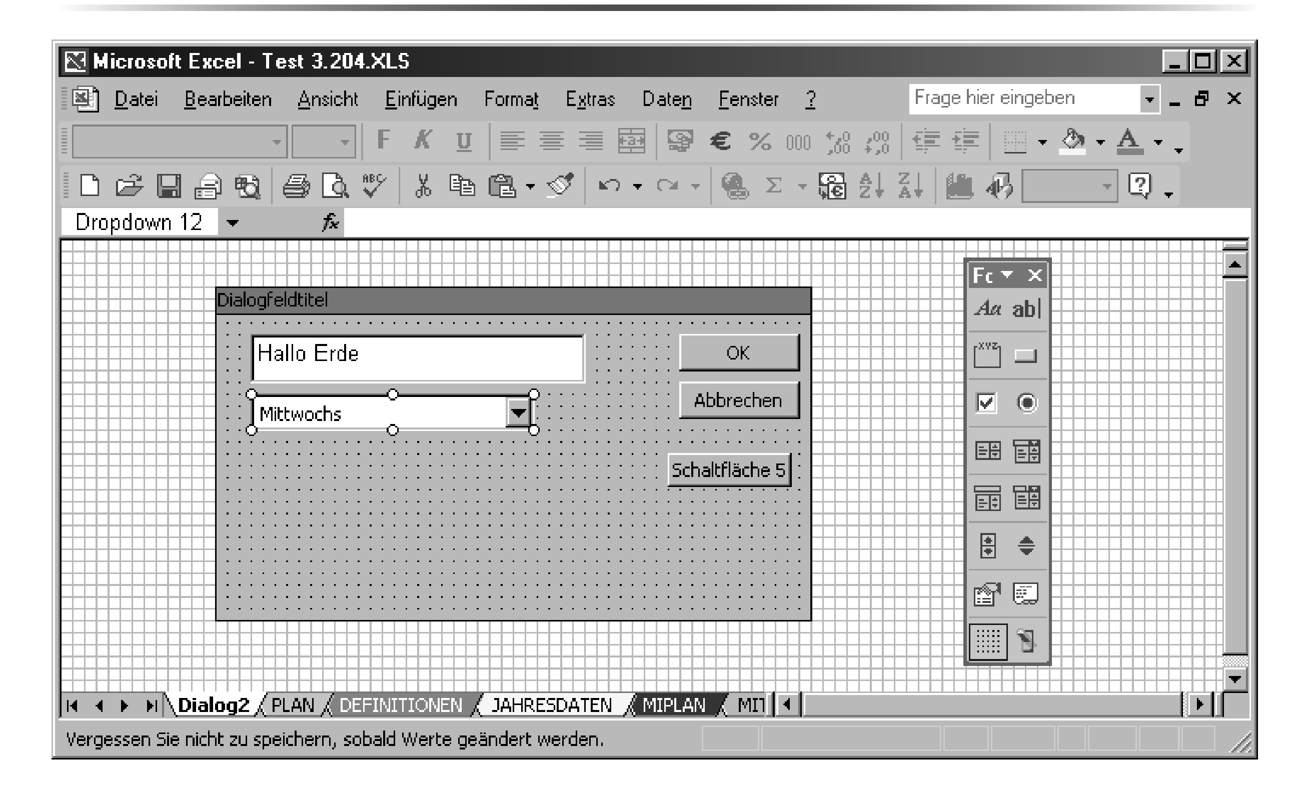
In der obigen Abbildung wirkt sich die Auswahl von „Optionsfeld 11“ nicht auf die Radio-Buttons innerhalb des Gruppenfeldes 6 aus. Optionsfelder können als Eigenschaften auch Werte aus dem Zellbereich der Tabelle beinhalten.

Um programmtechnisch auf ein Optionsfeld zuzugreifen ist die Eigenschaft .VALUE entscheident – und zwar im Sinne des booleschen Algebras.

```
MsgBox DialogSheets(„Dialog2“).[Optionsfeld 11].Value
```

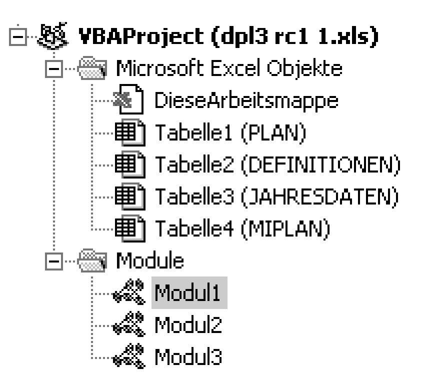
Wird obiger Befehl ausgeführt, mit gewähltem Optionsfeld 11, so wird der Wert 1 ausgegeben – bei nicht ausgewähltem Optionsfeld -4146. Um auf diese Werte entsprechend zu reagieren empfehlen sich Befehle in folgender Art:

```
a = DialogSheets(„Dialog2“).[Optionsfeld 11].Value
If a <> 1 Then MsgBox „Nicht gewählt.“ Else MsgBox „Gewählt.“
```



Das **Kombinationsfeld** „Dropdown" ist ein weiteres, klassisches Steuerungsfeld von Microsoft Windows. Über ein Rechtsklick lässt sich der Bereich auswählen, über den die Auswahlwerte entnommen werden. Ebenfalls über das Eigenschaftenmenü lässt sich bestimmen, in welche Zelle der Ausgabewerte übertragen werden soll. Im DPE3 werden die Ergebnisse stets direkt aus dem Fenster ausgelesen. Der Wert wird nicht durch den Inhalt, sondern durch eine Nummer repräsentiert, die mit einem Zellwert in Verbindung gebracht wird.

12 Kommentierter Quellcode



Der Quellcode dies Dienstplan in Excel gliedert sich in insgesamt sieben Teilen. Vier Teile für die jeweils sichtbaren Tabellen, wobei lediglich die Tabelle 1 – der Planer – über . Die ausgeblendeten Dialoge werde aus dem Modul heraus gesteuert.

Im Modul 1 finden sich die Befehle zum Ein- und Ausblenden der dritten Zeile, dem Neuaufbau des Plans, der Mitarbeiterplan und die Definition der Globalvariablen – Variablen die innerhalb des gesamten Dienstplan- Projekts verfügbar sind und der übergreifenden Kommunikation der einzelnen Programmteile dienen.

Das Modul 2 beherbergt den „CAC", die qualitative Planuntersuchung.

Im Modul 3 finden sich alle Programmteile, die von den Dialogfeldern benötigt werden.

Im Gegensatz zu den Kommentaren, die sich im Quellcode selbst finden, beginnend mit dem Hochkomma, sind die Kommentare, die in der größeren Schriftart des Buches verfasst sind Erklärungshilfen, die für den Gesamtkontext wichtig sind. Bevor der Leser versucht diesen Abschnitt des Kapitels durchzuarbeiten, sollten mindestens ein bis zwei Dienstpläne mit dem Dienstplan in Excel erstellt worden sein, da es die Sprache erklärt, die innerhalb des Quelltextes verwendet wird.

In seltenen Fällen finden sich drei aufeinander folgende Punkte (...), an diesen Stellen wurde der Quelltext gekürzt, da sich wiederholende Befehle nur durch kleine Parameter unterscheiden, die sich aus dem Zusammenhang ergeben. Bei der Vergabe von Variablennamen ging der Autor zuweilen wenig strategisch vor. In der tabellarischen Übersicht, kann aber der Wert nachgeschlagen werden.

12.1 Tabelle 1 – Plan

Der Befehl „Option Explicit" besagt, dass eine Deklaration von Variablen in jedem Fall erfolgen muss. Die Befehle CommandButton... mit dem CALL-Befehl verweisen in die Module – der Befehlsablauf wird dort fortgesetzt.

```
Private Sub CommandButton5_Click()
ActiveWorkbook.DialogSheets(„ZUSATZ").Show
End Sub
```

CommandButton5 bringt das ausgeblendete Dialogfeld „Zusatz" in den Vordergrund. Bis das Dialogfeld geschlossen wird, sei es durch einen Klick auf das X-Symbol oder durch die Auswahl eines Befehls, wird die Bedienung auf der Tabellenebene unterbunden. Nur am Rande: wird im obigen das .Show durch .Hide ersetzt würde das Dialogfeld wieder ausgeblendet.

```
Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)
```

geschrieben	STRING	Zu übergebender Text – wird zur Laufzeit zusammengefügt
ausgelesen	STRING	Zwischenpuffervariable
akt_row	VARIANT	Aktuelle Reihe auf dem Plan an der sich der Cursor befindet.
akt_col	VARIANT	Aktuelle Spalte auf dem Plan an der sich der Cursor befindet.
v_mitarbeitername	STRING*	Mitarbeitername wird aus dem Plan ausgelesen
v_kartennummer	STRING*	Kartennummer / Personalnummer wird auf dem Plan ausgelesen
v_mita_az	STRING*	Arbeitszeit : 1/1, 3/4, 1/2, ¼ oder WT
v_mita_an	STRING*	Dienstform z.B. KS oder SLV
Dienstart	BYTE*	Kategorie des Dienstes : Standard, Sonder- oder freie Dienstform
a	VARIANT	Laufvariable für Schleifen
Antwort	VARIANT	Nach Dialogfenster wird hier die gedrückte Schalfläche gespeichert
* = Globalvariablen für den Datenaustausch zwischen Dialog und Programm		

Folgender Programmabschnitt wird aufgerufen, sobald ein Doppelklick mit der linken Maustaste ausgelöst wird. Der Doppelklick hat, je nachdem an welcher Stelle er durchgeführt wird, unterschiedliche Funktionen.

Ein Doppelklick auf dem Namensfeld des Mitarbeiters öffnet beispielsweise das Dialogfeld „Mitarbeiter", während ein Doppelklick innerhalb des Planungsbreichs das Dialogfeld „Planungshelfer" einblendet. Zunächst wird mit den Befehlen:

```
akt_row = ActiveCell.Row
akt_col = ActiveCell.Column
```

festgestellt an welcher Stelle in der Tabelle sich die Schreibmarke befindet.

```
If akt_row < 9 Or akt_row > 73 Then Exit Sub
If akt_row Mod 3 <> 0 Then Exit Sub
If akt_col = 2 Or akt_col = 3 Then Exit Sub
If akt_col <> 1 Then GoTo eingabehelfer
```

Die Auswertung geschieht über eine Bereichsprüfung: Befindet sich die Schreibmarke außerhalb der Reihen 9 und 73, so wird der Programmablauf mit EXIT SUB sofort abgebrochen. Über eine Restwertbestimmung (Mod = Modulo Befehl) wird festgestellt, ob sich die Schreibmarke in der ersten Zeile eines Mitarbeiters befindet (muss sich glatt durch drei Teilen lassen, sprich der Befehl AKTUELLE_REIHE MODULO NULL darf keinen Restwert haben). Befindet sich die Schreibmarke in der Spalte 2 (B) oder 3 (C) so wird der Programmablauf ebenfalls abgebrochen. Ist die Spalte ungleich 1 so wird der „Planungshelfer" eingeblendet.

```
v_mitarbeitername = Cells(akt_row, akt_col).Value
v_kartennummer = Cells(akt_row + 2, akt_col).Value
ausgelesen = Cells(akt_row + 1, akt_col).Value
If Len(ausgelesen) > 6 Then
v_mita_az = Right$(ausgelesen, 3)
v_mita_an = Left$(ausgelesen, 3)
v_nachtwache = Cells(akt_row, 50).Value
End If
ActiveWorkbook.DialogSheets("MITARBEITER").Show
```

Bevor das Dialogfeld für die Mitarbeiterbearbeitung eingeblendet wird, müssen zunächst die Werte des aktuellen Mitarbeiters aus der Tabelle ausgelesen werden und in das Dialogfeld übertragen werden. Da der Doppelklick auf dem Namensfeld des Mitarbeiters aufgeführt wurde, kann einfach durch Addition zur jetzigen Position ein Wert angepeilt werden. Eine echte Wertprüfung findet hier nicht statt, lediglich ob der ausgelesene Wert länger als sechs Ziffern ist. Der Wert der Nachtwache findet sich in der Spalte AX (50) und wird ebenfalls in das Dialogfeld übertragen.


```
Cells(akt_row, akt_col).Value = v_mitarbeitername
Cells(akt_row + 2, akt_col).Value = v_kartennummer
Cells(akt_row + 1, akt_col).Value = v_mita_an & „ - „ & v_mita_az
Cells(akt_row, 50).Value = v_nachtwache
Exit Sub
```

Wird das Dialogfeld geschlossen, so wird der Programmablauf an dieser Stelle fortgesetzt und die neuen Werte werden in die Tabelle übertragen. Mit EXIT SUB endet anschließend der Programmablauf.

```
eingabehelfer:
,  Die Dienstart wird hier auf den Wert FÜNF gesetzt. Findet sich nach all den Pro-
   grammschritten keine Wertveränderung, so heisst dies - NICHTS eingegeben.
dienstart = 5
,  Einblendung des Dialogfelds - Programmablauf ist bis zum Klick von OK unterbrochen.
   ActiveWorkbook.DialogSheets(„PLANUNGSHELFER").Show
,  Es wurde keine Dienstform ausgewählt
If dienstart = 5 Then Exit Sub
```

Der Programmabschnitt EINGABEHELFER wurde weiter oben im Programmcode per GOTO-Befehl angesprungen. Bevor das Dialogfeld PLANUNGSHELFER eingeblendet wird, wird eine globale Variable auf den Wert 5 gesetzt. Ist auch nachdem das Dialogfeld geschlossen wurde, der Wert in DIENSTART immer noch auf 5, so wurde keine Änderung vorgenommen und der Programmablauf ohne Änderung an einem Wert beendet.

```
ActiveSheet.Unprotect
```

Bevor die Dienstart geändert wird, wird der Blattschutz der Excel-Tabelle mit dem obigen Befehl deaktiviert. Unabhängig wie der Programmverlauf weitergeführt wird, muss der Blattschutz später wieder aktiviert werden.

```
,  Es wurde ‚KEIN DIENST' gewählt
If dienstart = 4 Then
Cells(akt_row, akt_col).Value = „"
Cells(akt_row, akt_col).ClearComments
Cells(akt_row + 2, akt_col).Value = „"
'  Auch im Schattenplan wird ein möglicher Eintrag gelöscht.
Cells(akt_row, akt_col + 62).Value = „"
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
Exit Sub
End If
```

Wurde die Schaltfläche „KEIN DIENST" im Dialogfeld angeklickt, so kehrt wird der Wert DIENSTART auf 4 gesetzt und der oben genannte Programmverlauf sorgt für ein komplettes Entfernen aller Einträge zu diesem Dienst. Da Informationen für den Mitarbeiterplan im ausgeblendeten Schattenbereich der Excel-Tabelle abgelegt werden, müssen auch diese Einträge wieder entfernt werden.

```
,  Standardienstform wurde gewählt
If CByte(DialogSheets(„PLANUNGSHELFER").[Malparameter].Value) = 0 Then _
DialogSheets(„PLANUNGSHELFER").[Malparameter].Value = 1
If dienstart = 1 Then
Application.Cursor = xlWait ,Sanduhr
For a = 1 To Worksheets(„DEFINITIONEN").Cells(117 + _
CByte(DialogSheets("PLANUNGSHELFER").[Malparameter].Value), 3)
'  Prüfung, ob hier nicht in Tage geplant werden soll, die nicht zulässig sind
If akt_col + a - 1 > 34 - CInt(Range(„AO3").Value) Then
MsgBox "Uupps, das wäre beinahe ein Fehler geworden.", vbCritical, "NeeNee!"
Application.Cursor = xlDefault ,Normal
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
Exit Sub
End If
```

```
Cells(akt_row, akt_col + a - 1).Value = Worksheets(„DEFINITIONEN“).Cells(17 + _
CByte(DialogSheets(“PLANUNGSHELFER”).[Standarddienst].Value), 3)
Cells(akt_row, akt_col + a - 1).ClearComments
Cells(akt_row, akt_col + 62 + a - 1).Value = „“
If Cells(akt_row + 2, akt_col + a - 1).Value <> „“ Then
antwort = MsgBox(“Es wurde ein hinterlegter Zeitwert entdeckt, soll dieser entfernt “ & _
„werden ?“, vbYesNo, „Frage?“)
If antwort = „7“ Then Exit Sub
Cells(akt_row + 2, akt_col + a - 1).Value = „“
End If
Next a
Application.Cursor = xlDefault
,  Blattschutz wieder aktivieren
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
Exit Sub
End If
```

Es ist im Dialogfeld des Eingabehelfers möglich, die Wiederholung einer Dienstform einzugeben. Wird jedoch am 30ten des Monats eingegeben, man wolle fünfmal den gleichen Dienst, so würde dies in den nächsten Monat „hineinsprechen“. Um das zu verhindern wird überprüft, ob es denn überhaupt den „nächsten zulässigen Tag“ gibt.

Diese Prüfung wird mit der Zelle AO3 vorgenommen, die die Anzahl von Tagen beinhaltet, die dem aktuellen Monat auf die 31 fehlen.

Der Programmablauf endet hier. Der Mauszeiger wird auf den in Excel hinterlegten Normalwert zurückgesetzt. Der Blattschutz wird aktiviert, sodass der Bediener nicht versehentlich Änderungen vornehmen kann. Mit EXIT SUB wird der Programmablauf hier beendet,

```
,  Sonderdienstform wurde gewählt
If dienstart = 2 Then
Cells(akt_row, akt_col).ClearComments
If DialogSheets(“PLANUNGSHELFER”).[Sonderdienstform].Value = “10” Then Exit Sub
Cells(akt_row, akt_col).Value = Worksheets(“DEFINITIONEN”).Range(“C107”)
Cells(akt_row, akt_col + 62).Value = _
Worksheets(“DEFINITIONEN”).Cells(107 + _
CByte(DialogSheets(“PLANUNGSHELFER”).[Sonderdienstform].Value), 4)
Cells(akt_row + 2, akt_col).Value = Worksheets(„DEFINITIONEN“).Cells(107 + _
CByte(DialogSheets(“PLANUNGSHELFER”).[Sonderdienstform].Value), 3)
Cells(akt_row, akt_col).AddComment
Cells(akt_row, akt_col).Comment.Visible = False
geschrieben = Worksheets(“DEFINITIONEN”).Cells(107 + _
CByte(DialogSheets(“PLANUNGSHELFER”).[Sonderdienstform].Value), 4)
Cells(akt_row, akt_col).Comment.Text “ “ & geschrieben
End If
‘  Es wurde eine eigene, freie Definition ausgewählt
If dienstart = 3 Then
Cells(akt_row, akt_col).ClearComments
Cells(akt_row, akt_col).Value = Worksheets(„DEFINITIONEN“).Range(„C107“)
Cells(akt_row, akt_col + 62).Value = DialogSheets(„PLANUNGSHELFER“).[Titel].Caption
Cells(akt_row, akt_col).AddComment
Cells(akt_row, akt_col).Comment.Visible = False
geschrieben = DialogSheets(“PLANUNGSHELFER”).[Titel].Caption
Cells(akt_row, akt_col).Comment.Text “ “ & geschrieben
Cells(akt_row + 2, akt_col) = DialogSheets(“PLANUNGSHELFER”).[Zeitwert].Caption
End If
,  Blattschutz einschalten
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
End Sub
```


Wird eine Sonderdienstform gewählt, oder ein freier Dienst generiert, so wird in das Kommentarfeld der entsprechenden Zelle der Titel des Dienstes hinterlegt. Gleichzeitig werden diese Einträge in den Schattendienstplan (der Befehl ... akt_col + 62) eingetragen. Die Werte im Schattendienstplan werden für den Mitarbeiterplan benötigt.

WORKSHEETS_SELECTIONCHANGE

Private Sub Worksheet_SelectionChange(ByVal Target As Range)

Jede manuelle Veränderung durch den Benutzter führt dazu, dass der folgende Programmteil aktiviert wird. Bei automatisierten Zugriffen auf den Plan, beispielsweise aus einem Makro heraus, wird dieser Teil nicht ausgeführt.

geschrieben	STRING	Zu übergebender Text – wird zur Laufzeit zusammengefügt
ausgelesen	STRING	Zwischenpuffervariable
akt_row	VARIANT	Aktuelle Reihe auf dem Plan an der sich der Cursor befindet.
akt_col	VARIANT	Aktuelle Spalte auf dem Plan an der sich der Cursor befindet.
v_mitarbeitername	STRING*	Mitarbeitername wird aus dem Plan ausgelesen
v_kartennummer	STRING*	Kartennummer / Personalnummer wird auf dem Plan ausgelesen
v_mita_az	STRING*	Arbeitszeit : 1/1, 3/4, 1/2, ¼ oder WT
v_mita_an	STRING*	Dienstform z.B. KS oder SLV
Dienstart	BYTE*	Kategorie des Dienstes : Standard, Sonder- oder freie Dienstform
wert	VARIANT	Ausgelesener Stundenkontowert für Minus-Stunden-Berechnung
a	VARIANT	Laufvariable für Schleifen
Antwort	VARIANT	Nach Dialogfenster wird hier die gedrückte Schalfläche gespeichert
* = Globalvariablen – Datenaustausch zwischen Programm und Dialogfenster		

```
On Error GoTo marke2
, Aktuelle Position der akvtiven Zelle definieren
akt_row = ActiveCell.Row
akt_col = ActiveCell.Column
‘ Wurde eine Änderung außerhalb Spalte ‘B’ durchgeführt ?
If akt_col <> 2 Then GoTo pruefung2
If akt_row < 9 Or akt_row > 73 Then Exit Sub
If akt_row Mod 3 <> 1 Then Exit Sub
```

Da dieser Programmteil sehr oft aktiviert wird, muss hier schnell entschieden werden, ob es überhaupt eine Eingabe oder Änderung gibt, die den Aufruf rechtfertigen. Einerseits ist das der Fall, sobald eine Änderung am Stundenkonto vorgenommen wird. Da Excel zwar in der Lage ist Minusstunden zu errechnen, nicht jedoch welche durch den Benutzer eingeben zu lassen, muss hier auf einen kleinen Programmtrick zurückgegriffen werden. Anstelle das Stundenkonto (gelbes Feld auf der linken Seite) Excel gegenüber das Datumsfeld zu deklarieren, wird es als reines Textfeld angesprochen, so ist die Eingabe eines Wertes wie -01:50 überhaupt möglich. Damit Excel damit rechnen kann wird nun in einem Schattenbereich des Plans (Spalte CS ff) die Umrechnung aus dem Textfeld aus Spalte B vorgenommen.

```
wert = Cells(akt_row - 1, 2)
If wert = „“ Then wert = 0
If Left$(wert, 1) <> „-“ Then
Cells(akt_row - 1, 98).Value = wert
Else
Cells(akt_row - 1, 99).Value = Right$(wert, Len(wert) - 1)
Cells(akt_row - 1, 98).Value = Cells(akt_row - 1, 99).Value * -1
Cells(akt_row - 1, 99) = 0
End If
Exit Sub
```

Neben der Stundenkontoberechnung ist lediglich die Eingabe des Schlüsselworts für die Teamsitzung ein Vorgang, der vom Programm abgefangen werden müsste (im Bereich pruefung2).

```
,   TEAMSITZUNG
pruefung2:
,   Variablen einlesen
team_key = Worksheets(„DEFINITIONEN“).Cells(95, 3).Value
team_as = Worksheets(„DEFINITIONEN“).Cells(96, 3).Value
team_wert = Worksheets(„DEFINITIONEN“).Cells(97, 3).Value

If Cells(akt_row - 1, akt_col).Value = team_key And Range(„AR4“).Value = 0 Then
‘ Sicherheitsabfrage
antwort = MsgBox(“Gemäß den Definitionsparametern kann nun die Teamsitzung im Plan
“ & _
“ eingetragen werden. In diesem Rahmen werden alle nicht als Dienst geplanten “ & _
“Mitarbeiter zur Teamsitzung einbestellt und der hinterlegte Stundenwert eingetragen.”
& _
“Evtl. vorhandene Stundenwerte in der dritten Berechnungszeile werden in diesem “ & _
“Rahmen überschrieben. Soll der Vorgang fortgesetzt werden ?”, vbYesNo, “Frage!”)
If antwort = 7 Then Exit Sub
‘ Blattschutz abschalten
ActiveSheet.Unprotect
, Teamsitzungswerte eintragen
Application.Cursor = xlWait ,Sanduhr
For laufvariable = 9 To 72 Step 3
‘ Spätdienstreglement
If Worksheets(“DEFINITIONEN”).Cells(99, 3).Value >= 1 And _
Cells(laufvariable, akt_col).Value = Worksheets(“DEFINITIONEN”).Cells(20, 3).Value _
Then GoTo sprungmarke
‘ Frühdienstreglement I
If Worksheets(“DEFINITIONEN”).Cells(99, 3).Value = 2 And _
Cells(laufvariable, akt_col).Value = Worksheets(“DEFINITIONEN”).Cells(18, 3).Value _
Then GoTo sprungmarke
‘ Frühdienstreglement I
If Worksheets(“DEFINITIONEN”).Cells(99, 3).Value = 2 And _
Cells(laufvariable, akt_col).Value = Worksheets(“DEFINITIONEN”).Cells(19, 3).Value _
Then GoTo sprungmarke
‘ Urlauber filtern
If Cells(laufvariable, akt_col).Value = Worksheets(„DEFINITIONEN“).Cells(24, 3).Value _
Then GoTo sprungmarke
, Nachtwächterreglement
If Worksheets(„DEFINITIONEN“).Cells(100, 3).Value = 0 And _
Cells(laufvariable, akt_col).Value = Worksheets(„DEFINITIONEN“).Cells(23, 3).Value _
Then GoTo sprungmarke
‘ Wenn kein Name oder Name LEER ist überspringen
If Cells(laufvariable, 1).Value = „“ Or Cells(laufvariable, 1) = „Leer“ Then GoTo _
sprungmarke
‘ Prüfen ob Krankenpflegeschüler in die TS kommen
If Left$(Cells(laufvariable + 1, 1).Value, 3) = „KPS“ And _
Worksheets(„DEFINITIONEN“).Cells(101, 3).Value = 0 Then GoTo sprungmarke
‘ Prüfen ob ZDL in die TS kommen
If Left$(Cells(laufvariable + 1, 1).Value, 3) = „ZDL“ And _
Worksheets(„DEFINITIONEN“).Cells(102, 3).Value = 0 Then GoTo sprungmarke

Cells(laufvariable, akt_col).Select
Cells(laufvariable + 1, akt_col).Value = team_as
Cells(laufvariable + 2, akt_col).Value = team_wert

If Worksheets(“DEFINITIONEN”).Cells(98, 3).Value = 1 Then
With Selection.Interior
.Pattern = xlGray8
```



```
.   PatternColorIndex = 17
End With
End If
sprungmarke:
Next laufvariable
‘   Teamsitzungsmerker auf 1 setzen
Range(“AR4”).Value = 1
‘   Zelle unterhalb der Eingabezelle selektieren
Cells(akt_row + 1, akt_col).Select
,   Blattschutz wieder aktivieren
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
End If
marke2:
Application.Cursor = xlDefault ,Normal
End Sub
```

Da es für eine Teamsitzung verschiedenste Formen der Durchführung gibt, finden sich hier sehr viele IF-Abfragen, um eine möglichst flexible Darstellung zu ermöglichen. Prozedural betrachtet beginnt der Ablauf mit der Feststellung welches das Schlüsselwort für die Teamsitzung in der Zeile 8 erwartet wird. Findet sich dieses Schlüsselwort so wird quasi der komplette Programmblock als IF-Bedingung durchgeführt, die IF-Abfrage endet erst nachdem der Blattschutz im unteren Programmbereich wieder aktiviert wird.

Ansonsten erklärt sich der Programmverlauf aus den Möglichkeiten, die über die DEFINTIONS-Tabelle geboten werden. Nachteilig ist die fehlende Umsetzung des Auslesens der Namensfelder direkt in den VBA-Quellcode. In der vorliegenden Fassung ist die Umsortierung von Definitionen nur mit den jeweiligen Änderungen im Programmcode möglich.

12.2 Modul 1

```
Option Explicit
Public gewaehlter_monat As String
Public durchlauf
Public berechnungsdurchlauf
Public v_mitarbeitername As String
Public v_mita_az As String
Public v_kartennummer As String
Public v_mita_an As String
Public team_key As String
Public team_as As String
Public team_wert As String
Public dienstart As Byte
Public v_nachtwache As String
```

Im Modul 1 finden sich zunächst die Definitionen für die globalen Variablen, die innerhalb der gesamten Excel-Tabelle und allen Programmabschnitten genutzt werden können.

```
Sub Ausblendung()
Dim wert As Variant
Dim lv As Integer
ActiveSheet.Unprotect
For lv = 11 To 73 Step 3
wert = CStr(lv) & „:“ & CStr(lv)
Worksheets(„PLAN“).Rows(wert).EntireRow.Hidden = True
Next lv
Worksheets(„PLAN“).Rows(„1:3“).EntireRow.Hidden = True
Range(„D9“).Select
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
End Sub
```

Der Befehl „Ausblendung“ sorgt dafür, dass die dritte Zeile mit den einzelnen Stundenwerten komplett ausgeblendet wird. Hintergrund ist die bessere Übersicht bei Monitoren unterhalb der Auflösung 1024x768 Pixel.

```
Sub Einblendung()
, Mit V2.00 eingeführt
, Dieser Befehl hebt den Befehl ‚Ausblendung‘ wieder auf. Hintergrund ist die Aus-
, blendung der DRITTEN ZEILE. Das soll die Übersicht verbessern.
ActiveSheet.Unprotect
Rows(„1:73“).Select
Selection.EntireRow.Hidden = False
Range(„D9“).Select
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
End Sub
```

Als Gegenstück zur „Ausblendung“ fungiert „Einblendung“.

PLAN-KILLER:

Der größte Programmabschnitt im gesamten DPE3-Projekt ist der scherzhaft benannte Plan-Killer.

monatsname	STRING	Ausgelesener Name des neuen Monat
monatspara	STRING	Ausgelesener Sollstundenwert des neuen Monat
Antwort	STRING	Nach Dialogfenster wird hier die gedrückte Schalfläche gespeichert
laufvariable	INTEGER	Laufvariable für Hauptschleifen
lese	VARIANT	Ausgelesener Wert aus der Zelle
vorkomma	VARIANT	Anteil vor dem Komma für die Stundenübernahme
nachkomma	VARIANT	Anteil nach dem Komma für die Stundenübernahme
wert	VARIANT	Ausgelesener Wert aus der Zelle 1
auslesewert	VARIANT	Ausgelesener Wert aus der Zelle 2
ausdemplan	VARIANT	Ausgelesener Wert aus der Zelle 3
ausfeiertage	VARIANT	Ausgelesener Wert aus der Zelle 4
vk$	STRING	Anteil vor dem Komma für die Stundenübernahme als String
nk$	STRING	Anteil nach dem Komma für die Stundenübernahme als String
a	INTEGER	Laufvariable für Tage des Monats
b	INTEGER	Laufvariable für Prüfung auf Feiertage
gewaehlter_monat	STRING*	Aus dem Dialogfeld übergebener Wert der ausgewählten Monat
* = Globalvariable		

Die Aufgabe des Plan-Killers ist die Bereitstellung eines neuen Monatsplanes. Funktionell unterteilt sich der Programmabschnitt in mehrere Segmente, wobei der größte Teil mit dem Makrorekorder aufgezeichnet wurde und Inhalte des alten Plans löscht. Bevor der Befehl ausgeführt wird, muss die vorherige Planversion in jedem Fall gesichert werden.

```
,   *** Sicherheitsabfrage
    antwort = MsgBox(„Sind Sie sicher dass alle Werte des Planes gelöscht werden sol-
    len „ & _ „          um einen neuen Monatsplan aufzubauen ? (Die Parameter der Mit
    arbeiter bleiben „ & _ „erhalten)“, vbYesNo, „Frage!“)
If antwort = 7 Then Exit Sub
,   *** Mit V3.00 eingeführt
,   Soll der Kontowert übernommen werden ?
antwort = MsgBox(„Sollen die Kontodaten, wie auf diesem Plan angezeigt, als Stunden-
    konto“ & _ „auf den neu zu erstellenden Plan übernommen werden? Wenn Sie auf
    NEIN klicken werden die“ & _ „Altdaten mit 00:00 überschrieben. Klicken Sie auf AB-
    BRECHEN um den Vorgang abzubrechen.“, vbYesNoCancel, „Frage!“)
,   *** Mit V3.00 eingeführt
,   Es wurde ‚ABBRECHEN‘ geklickt - sprich der Vorgang soll gar nicht durchgeführt werden.
```



```
If antwort = 2 Then Exit Sub
```

Zunächst eine Sicherheitsabfrage, ob mit dem Neuaufbau des Plans überhaupt begonnen werden soll. Anschließend eine Abfrage die feststellt, ob die Stundenwerte des jetzigen Plans als Stundenkontowert des zukünftigen Plan genutzt werden sollen.

```
If antwort = 6 Then
For laufvariable = 9 To 72 Step 3
lese = CDbl(Cells(laufvariable, 37).Value * 24)
vorkomma = Fix(lese)
vk$ = CStr(vorkomma)
If vorkomma < 1 Then
nachkomma = Round((lese - vorkomma) * -60, 2)
Else
nachkomma = Round((lese - vorkomma) * 60, 2)
End If
If nachkomma >= 10 Then
nk$ = CStr(Fix(nachkomma))
Else
nk$ = “0” & CStr(Fix(nachkomma))
End If
Cells(laufvariable, 2) = vk$ & „:“ & nk
‘ Berechnung etwaiger Minusstunden
wert = Cells(laufvariable, 2)
If wert = „“ Then wert = 0
If Left$(wert, 1) <> „-“ Then
Cells(laufvariable, 98).Value = wert
Else
Cells(laufvariable, 99).Value = Right$(wert, Len(wert) - 1)
Cells(laufvariable, 98).Value = Cells(laufvariable, 99).Value * -1
Cells(laufvariable, 99) = 0
End If
Next laufvariable
End If
```

Der Bereich „Stundenübernahme“ ist als IF-Block realisiert und besitzt einige Anteile aus dem Minusstunden-Berechner aus Tabelle 1.

```
, Nachtwachen-Merker durchlaufen lassen
For laufvariable = 10 To 73 Step 3
auslesewert = Cells(laufvariable, 2).Value
Cells(laufvariable - 1, 50).Value = auslesewert
Next laufvariable
‘ Es wurde NEIN geklickt - so werden alle Stundenkontenwerte auf 00:00 gesetzt
If antwort = 7 Then
For laufvariable = 9 To 72 Step 3
Cells(laufvariable, 2) = „00:00“
Cells(laufvariable, 98).Value = 0
Next laufvariable
End If
```

Die Übernahme der Nachtwachen und die Prüfung, ob die Stundenkontowerte tatsächlich eingetragen werden sollen, geschieht über obigen Programm-Abschnitt. Da hier ein Ablauf in jedem Fall durchgeführt wird, ist einiges an Potential für die Performance-Verbesserung vorhanden.

```
,   *** Blattschutz aufheben
ActiveSheet.Unprotect
,   *** Alle Sperrungen aufheben
Range(„AE9:AH74“).Select
    Selection.Locked = False
Range(„AE5:AH7“).Font.Color = vbBlack
‘   *** Einträge des Monats entfernen
```

```
‘   Die Einträge sehen deshalb so langatmig aus, da sie mit dem Makrorekorder mit-
‘   geschrieben wurden. Es werden in allen folgenden Befehlen nichts anderes als die
,   Leerwerte wieder hergestellt. Etwaige Schattenpläne werden zurückgesetzt. Team-
,   sitzungseinträge und Kommentare entfernt.
    Range(„D9:AH74“).Select
    Selection.ClearContents
    ActiveWindow.LargeScroll Down:=-1
    Range("C10,C13,C16,C19,C22,C25,C28,C31,C34,C37,C40,C43,C46,C49,C52").
Select
    Range("C52").Activate
    ActiveWindow.ScrollRow = 9
    {...}
    Range( _
    "C10,C13,C16,C19,C22,C25,C28,C31,C34,C37,C40,C43,C46,C49,C52,C55,C58,
    C61,C64,C67,C70,C73" _
    ).Select
    Range("C73").Activate
    Selection.ClearContents
    Selection.ClearComments
    ActiveWindow.ScrollRow = 32
    ActiveWindow.ScrollRow = 31
    ActiveWindow.ScrollRow = 28
    ActiveWindow.ScrollRow = 25
    ActiveWindow.ScrollRow = 20
    ActiveWindow.ScrollRow = 15
    ActiveWindow.ScrollRow = 8
      Range("B9,B12,B15,B18,B21,B24,B27,B30,B33,B36,B39,B42,B45,B48,B51,B54").
Select
    Range("B54").Activate
    ActiveWindow.ScrollRow = 9
    {...}
    Range( _
    "B9,B12,B15,B18,B21,B24,B27,B30,B33,B36,B39,B42,B45,B48,B51,B54,B57,B60,
    B63,B66,B69,B72" _
    ).Select
    Range("B72").Activate
    ActiveCell.FormulaR1C1 = "00:00"

    Range( _
    "B9,B12,B15,B18,B21,B24,B27,B30,B33,B36,B39,B42,B45,B48,B51,B54,B57,B60,B63,B66,
    B69,B72" _
    ).Select
    ActiveWindow.ScrollRow = 8
    Range("D7:AH7").Select
    Selection.ClearContents
    Selection.ClearComments
    Range("D8:AH8").Select
    Selection.ClearContents
    Range( _
"AI9,AI12,AI15,AI18,AI21,AI24,AI27,AI30,AI33,AI36,AI39,AI42,AI45,AI48,AI51,AI54,AI57,AI60,AI63,AI
66,AI69,AI72" _
    ).Select
    Range("AI72").Activate
    Selection.ClearContents
```

Der gesamte Abschnitt wurde mit dem Makrorekorder aufgezeichnet und entfernt lediglich Daten und Formatierungen aus dem alten Plan.

```
ActiveWorkbook.DialogSheets(„Dialog1“).Show
```

Das Dialogfeld „Dialog1“ wird eingeblendet und liefert über die Globalvariable „gewaehlter_monat“ den

Wert zurück, was für ein Plan aufgebaut werden soll. Es findet sich also keine Beschränkung immer nur den nächsten Monat zu generieren.

```
If gewaehlter_monat <= 0 Then Exit Sub
If gewaehlter_monat >= 18 Then Exit Sub
```

Beide Befehle dienen nur als kleine Sicherung, da es sonst keine Prüfung gibt, ob der Wert korrekt zurückgegeben wird.

```
,   Aus der Tabelle JAHRESATEN wird der Monat und das Stundensoll übernommen und
,   in die beiden Variablen monatsname und monatspara übergeben.
monatsname = Worksheets(„JAHRESDATEN“).Cells(1 + gewaehlter_monat, 2).Value
monatspara = Worksheets(„JAHRESDATEN“).Cells(1 + gewaehlter_monat, 4).Value
,   Der Aktuelle Monat wird als DATUMSFORMAT in die Zelle I9 eingetragen
Cells(1, 9) = monatsname
,   Aus der Definitionstabelle JAHRESDATEN wird nun der Sollstundenwerte ausgelesen
,   und auf die aktuelle Tabelle übertragen. Der wert gewaehlter_monat entstammt aus
,   dem Dialogfeld, welches zum jetzigen Zeitpunkt bereits wieder geschlossen ist.
Sheets(„JAHRESDATEN“).Select
Cells(1 + gewaehlter_monat, 3).Select
Selection.Copy
Sheets(„PLAN“).Select
Cells(2, 22).Select
ActiveSheet.Paste
```

Der neue Plan wird aus den Werten der Tabelle JAHRESDATEN aufbereitet.

```
,   Prüfung des Quartals , wenn das nun 1 ist, wird Wachenblock zurückgesetzt
If Worksheets(„JAHRESDATEN“).Cells(1 + gewaehlter_monat, 5).Value = 1 Then
Range(„AX9,AX12,AX15,AX18,AX21,AX24,AX27,AX30,AX33,AX36,AX39,AX42,AX45,AX48,
AX51,AX54,AX57,AX60,AX63,AX66,AX69,AX72“).Select
Selection.Value = 0
End If
```

Über den Marker 1 bei in der Tabelle JAHRESDATEN gibt es die Möglichkeit, das Mitzählen der Nachtwachen abzubrechen. Soll beispielsweise niemals mitgezählt werden, so ist hinter jedem Monatseintrag in JAHRESDATEN eine 1 zu schreiben.

```
,   Datumsleiste wird aufgebaut. Die Befehle sind mit dem Makrorekorder mitgeschrieben
,   worden und sind wie folgt zustande gekommen. In die erste Zelle (Tag 1 des neuen
,   Monats) wird das Datum aus der Variable monatspara eingetragen und dann entsprechend
,   mit dem Auffüllkästchen nach rechts Richtung 31ten gezogen. Kurzzeitig werden immer
,   31 Tage eingetragen. Eine andere Funktion wird „überhängende“ Tage des Folgemonats
,   abschneiden - dafür wird wieder auf eine Excel-Funktionalität zurückgegeriffen.
Range(„D5“).Select
ActiveCell.FormulaR1C1 = monatspara
Range(„D5“).Select
Selection.AutoFill Destination:=Range(„D5:AH5“), Type:=xlFillDefault
Range(„D5:AH5“).Select
Range(„AG6“).Select
Selection.AutoFill Destination:=Range(„AG6:AH6“), Type:=xlFillDefault
Range(„AG6:AH6“).Select
Range(„D9“).Select
```

Aufbau der Datumsleiste des neuen Plans, hinter dem Kommentarzeichen steht eigentlich alles, was es zu diesem Abschnitt zu sagen gibt.

```
, *** Feiertage aufbauen
For a = 4 To 34
ausdemplan = Cells(5, a).Value
For b = 20 To 49
```

```
ausfeiertage = Sheets(„JAHRESDATEN“).Cells(b, 2).Value
If ausdemplan = ausfeiertage Then
Cells(7, a).Value = „ft“
Cells(7, a).Select
Selection.Font.Bold = True
Selection.Font.ColorIndex = 7
Cells(7, a).AddComment
Cells(7, a).Comment.Visible = False
Cells(7, a).Comment.Text Text:=Sheets(„JAHRESDATEN“).Cells(b, 3).Value
End If
Next b
Next a
```

Vergleichend werden nun die aktuelle neue Tabelle mit den Einträgen in der Tabelle JAHRESDATEN verglichen, auf der Suche nach einem gefundenen Feiertag, der über diesen Abschnitt eingetragen wird. Der Name des Feiertags wird als Kommentarfeld eingetragen.

```
,   *** Nicht zulässige Monatstage am Ende deaktivieren
If Cells(3, 42).Value = 1 Then
Range(„AE9:AE74“).Select
Selection.Locked = True
Range(„AE5:AE7“).Font.Color = vbWhite
End If
If Cells(3, 43).Value = 1 Then
Range(„AF9:AF74“).Select
Selection.Locked = True
Range(„AF5:AF7“).Font.Color = vbWhite
End If
If Cells(3, 44).Value = 1 Then
Range(„AG9:AG74“).Select
Selection.Locked = True
Range(„AG5:AG7“).Font.Color = vbWhite
End If
If Cells(3, 45).Value = 1 Then
Range(„AH9:AH74“).Select
Selection.Locked = True
Range(„AH5:AH7“).Font.Color = vbWhite
End If
```

Hat ein Monat weniger als 31 Tage, so müssen „überlange“ Tage aus dem Plan entfernt werden. Um eine Eingabe zu verhindern, werden mit Hilfe des Codes die ganzen Spalten gegenüber Einträge mit einem SELECTION.LOCK geschützt.

```
Springweg:
,   *** Hintergrund wieder WEISS
Range(„D9:AH72“).Select
Selection.Interior.ColorIndex = xlNone
,   *** Teamsitzung zurücksetzen
Range(„AR4“).Value = 0
,   *** Sonderdienstkopieeinträge zurücksetzen
ActiveWindow.SmallScroll Down:=-3
    Range( _

„  BN9:CR9,BN12:CR12,BN15:CR15,BN18:CR18,BN21:CR21,BN24:CR24,BN27:CR27,
   BN30:CR30,BN33:CR33,BN36:CR36,BN39:CR39,BN42:CR42,BN45:CR45,BN48:
CR48“ _
    ).Select
    Range(„BN48“).Activate
    ActiveWindow.ScrollColumn = 68
        {...}
    ActiveWindow.ScrollColumn = 50
```



```
    Range( _
„BN9:CR9,BN12:CR12,BN15:CR15,BN18:CR18,BN21:CR21,BN24:CR24,BN27:
CR27,BN30:CR30,BN33:CR33,BN36:CR36,BN39:CR39,BN42:CR42,BN45:CR45,BN48:
CR48,BN51:CR51“ _
    ).Select
    Range(„BN51“).Activate
    ActiveWindow.SmallScroll Down:=17
    Range( _

„BN9:CR9,BN12:CR12,BN15:CR15,BN18:CR18,BN21:CR21,BN24:CR24,BN27:
CR27,BN30:CR30,BN33:CR33,BN36:CR36,BN39:CR39,BN42:CR42,BN45:CR45,BN48:
CR48,BN51:CR51,BN54:CR54,BN57:CR57,BN60:CR60,BN63:CR63,BN66:CR66,BN69:
CR69“ _
    ).Select
    Range(„BN69“).Activate
    ActiveWindow.SmallScroll Down:=12
    Range( _

„BN9:CR9,BN12:CR12,BN15:CR15,BN18:CR18,BN21:CR21,BN24:CR24,BN27:
CR27,BN30:CR30,BN33:CR33,BN36:CR36,BN39:CR39,BN42:CR42,BN45:CR45,BN48:
CR48,BN51:CR51,BN54:CR54,BN57:CR57,BN60:CR60,BN63:CR63,BN66:CR66,BN69:
CR69,BN72:CR72“ _
    ).Select
    Range(„BN72“).Activate
    Selection.ClearContents
    ActiveWindow.SmallScroll Down:=-30

‘   Kommentare entfernen
 Range(“D9:AH74”).Select
Selection.ClearComments
‘   *** Schutz des Wachenstatus wieder einschalten
Range(“B10,B13,B16,B19,B22,B25,B28,B31,B34,B37,B40,B43,B46,B49,B52”).Select
    Range(“B52”).Activate
    ActiveWindow.ScrollRow = 18
    Range( _

“B10,B13,B16,B19,B22,B25,B28,B31,B34,B37,B40,B43,B46,B49,B52,B55,B58,B61,B
64,B67” _
        ).Select
    Range(“B67”).Activate
    ActiveWindow.ScrollRow = 29
    Range( _

“B10,B13,B16,B19,B22,B25,B28,B31,B34,B37,B40,B43,B46,B49,B52,B55,B58,B61,B
64,B67,B70,B73” _
        ).Select
    Range(“B73”).Activate
    Selection.Locked = True
    Selection.FormulaHidden = False
‘   *** Blattschutz wird wieder aktiviert
On Error Resume Next
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
Range(„D9“).Select
Beep
Application.Cursor = xlDefault
Exit Sub
```

Mit dem obigen Abschnitt wird der Neuaufbau des Plans abgeschlossen.

```
fehlermeldung:
MsgBox „Bei der Übertragung ist ein Fehler aufgetreten. Prüfen Sie, ob die Stundenkontowerte“ & _
„gültige Werte im Sinne von [VORZEICHEN STUNDEN (max.3 stellig) : MINUTEN (max.2 stellig“ & _
„beinhalten.“, vbCritical, „Fehler!“
End Sub
```

Der Fehlerhandler für den „Plan-Killer“ besteht lediglich aus obiger Fehlermeldung. In einer künftigen Version soll in den „Plan-Killer“ eine stärkere Selbstreparaturfähigkeit des Planers eingebaut werden, indem selbst Befehlsbereiche wiederaufgebaut werden.

MIPLAN

Der Mitarbeiterplan ist eine Möglichkeit die Darstellung eines Dienstplans alternativ für den jeweiligen Mitarbeiter optimiert darzustellen. Die Ausgabe erfolgt über die Tabelle „MIPLAN“, die dynamisch verändert wird. Performance-seitig wäre die Einbindung von Microsoft Word sicherlich zügiger, doch würde das gegen den Grundsatz mit EINER DATEI = EIN PLAN verstoßen.

reg(6)	STRING	Zeiträume der Dienstformen für reguläre Dienste.
wot(6)	STRING	Zeiträume der Dienstformen für Wochenturnus.
dfo(6)	STRING	Bezeichner der Dienstarten in einem Array.
schnipp	STRING	Erster Teil des ausgeschnittenen Zeitwerts
schnapp	STRING	Hinterer Teil des ausgeschnittenen Zeitwerts
untersuchung	STRING	Wird für die Suche nach dem Doppelpunkt bei den Zeitwerten verwendet.
zeile_zwei	STRING	Wird für die Suche nach dem Bereitschaftsdienst-Symbol benötigt.
bereitsch.	STRING	Ausgelesener Wert aus der DEFINITION, was ist das Symbol für den Bereitschaftsdienst.
sonderdienst	STRING	Ausgelesener Wert aus der DEFINITION, was ist das Symbol für den Bereitschaftsdienst.
mitarbeitername	STRING	Name des Mitarbeiters
antwort	STRING	Rückgabewert aus der Sicherheitsabfrage
dienstform	STRING	Ausgelesener Dienst aus dem Plan
zweite_zeile	STRING	Ausgelesene zweite Zeile aus dem Plan
mitdienst	STRING	Mit wem hat der Mitarbeiter Dienst ?
dienstform	STRING	Aktueller Dienst des Mitarbeiters
dienstzeit_anzeigen	BYTE	Ausgelesener Wert aus der DEFINITION, ob die Anzeige der Dienstzeit überhaupt durchgeführt werden soll.
bereitschaft	BYTE	Ausgelesener Wert aus der DEFINITION, ob der Bereitschaftsdienst ausglesen werden soll.
dienstdauer	VARIANT	Zeitwert der Teamsitzung
laufvariable	INTEGER	Freie Laufvariable
laeufer	INTEGER	Freie Laufvariable 2
Tag	INTEGER	Schleifenvariable für den Monatstag
a	INTEGER	Schleifenvariable für Aufbau der Arrays
ilauf	INTEGER	Interne Laufvariable zur Feststellung, mit wem der aktuelle Mitarbeiter (laufvariable) Dienst hat.

```
antwort = MsgBox(„MiPlan für alle Mitarbeiter auf dem Drucker ausgeben ? Dieser“ & _
„ Vorgang kann ein Weilchen dauern.“, vbYesNo, „Frage!“)
If antwort = 7 Then Exit Sub
Application.Cursor = xlWait
Application.Calculation = xlCalculationManual
Application.ScreenUpdating = False
```


Der Programmverlauf beginnt zunächst mit einer Sicherheitsabfrage, ob die Ausgabe des Mitarbeiterplans wirklich beginnen soll. Vor der Einführung der Befehle Application. Calculation = xlCalculation Manual und Application.ScreenUpdating = False war die Ablaufgeschwindigkeit deutlich geringer. Am Ende einer jeden Prozedur, in der die Befehle zu Geschwindigkeitssteigerung verwendet werden, muss eine Neukalkulation des Excel-Tabellenblatts und die Ansicht wieder eingeschaltet werden.

```
dienstzeit_anzeigen = Worksheets(„DEFINITIONEN“).Range(„C103“).Value
bereitschaft = Worksheets(„DEFINITIONEN“).Range(„C105“).Value
bereitschaftssymbol = Worksheets(„DEFINITIONEN“).Range(„C106“).Value
sonderdienst = Worksheets(„DEFINITIONEN“).Range(„C107“).Value
{...}
Worksheets(„MIPLAN“).Range(„C4:E34“).ClearContents
Worksheets(„MIPLAN“).Range(„C1“).Value = Range(„I1“).Value
```

Alle Stammwerte, die für den Programmablauf benötigt werden, werden hier zu Beginn eingelesen. Technisch ist es der Plan so gelöst, das eine gesamte Tabelle in Microsoft Excel für die Ausgabe mit den individuellen Daten des jeweiligen Mitarbei
ters versorgt wird. Zunächst wird der Inhalt mit .CLEARCONTENTS gelöscht und der Monatsname übergeben.

```
If Worksheets(„MIPLAN“).Cells(4, 2).Value <> Cells(5, 4).Value Then
For laufvariable = 4 To 35
Worksheets(„MIPLAN“).Cells(laufvariable, 2).Value = Cells(5, laufvariable).Value
Next laufvariable
End If
```

Um ein wenig Zeit zu sparen prüft der obige Abschnitt, ob eventuell der Datumsbereich für den auszudruckenden Monat bereitgestellt wurde. Ist das der Fall, so kann die Kopie aus dem Planer übersprungen werden.

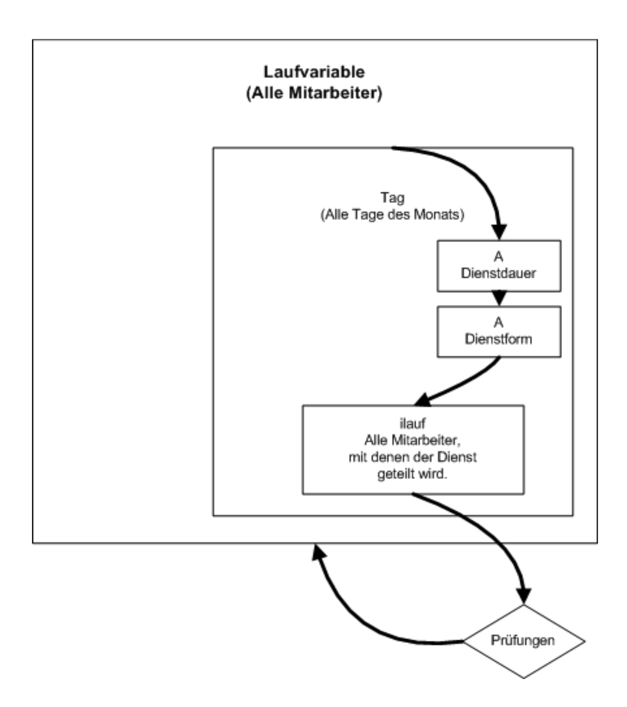
```
For laufvariable = 1 To 6
dfo(laufvariable) = Worksheets(„DEFINITIONEN“).Cells(17 + laufvariable, 3).Value
Next laufvariable
{...}
Next laufvariable
```

Alle flexiblen Dienstformen (Frühdienst, Spätdienst etc.) werden in ein kleines Array eingelesen, ebenso die zeitliche Dauer der Dienste einmal für Standarddienste (reg) und für Mitarbeiter im Wochenturnus (wot).

```
For laufvariable = 9 To 72 Step 3

mitarbeitername = Cells(laufvariable, 1)
If mitarbeitername = “” Or mitarbeitername = “Leer” Then GoTo sprungmarke
Worksheets(„MIPLAN“).Range(„C2“).Value = mitarbeitername
Worksheets(„MIPLAN“).Range(„F1“).Value = Cells(laufvariable, 36).Value
Worksheets(„MIPLAN“).Range(„F2“).Value = Cells(laufvariable, 3).Value
```

Die erste Schleife mit „laufvariable” bildet die Hauptschleife über alle 22 Mitarbeiter hinweg. Hat der aktuelle Mitarbeiter keinen Namen oder den Namen „Leer” so springt das Programm direkt an die Sprungmarke, die einen NEXT auslöst. So wird verhindert, dass leere Dienstpläne ausgegeben werden, für Mitarbeiter die nicht existieren. Die drei WORKSHEET-Befehle übertragen den Namen, Sollstunden- und Ist-Stundenwert auf den Mitarbeiterplan. Die komplette Struktur der Schleifen ist aus der Abbildung zu entnehmen.




```
For Tag = 4 To 35
dienstform = Cells(laufvariable, Tag).Value
zeile_zwei = Cells(laufvariable + 1, Tag).Value
If dienstzeit_anzeigen = 1 Then
If UCase$(Right$(Cells(laufvariable + 1, 1), 2)) = „WT“ Then
For a = 1 To 6
If dfo(a) = dienstform Then dienstdauer = wot(a)
Next a
Else
For a = 1 To 6
If dfo(a) = dienstform Then dienstdauer = reg(a)
Next a
End If , If von ‚WT‘ beendet
End If ‘ If von Dienstzeit anzeigen beendet
```

Dieser Programmabschnitt analysiert den Dienst des Mitarbeiters an dem jeweiligen Tag und liest die Dienstdauer (von … bis) für den Mitarbeiterplan aus. Eine Unterscheidung nach regulärer Dienstart und einem Wochenturnusmitarbeiter wird vorgenommen.

```
For ilauf = 9 To 72 Step 3

If Cells(ilauf, Tag).Value = „U“ Or Cells(ilauf, Tag) = „K“ Or _
Cells(ilauf, Tag) = “” Or Cells(ilauf, Tag) = sonderdienst _
Then GoTo isprung
```

Nun gilt es festzustellen, welche Kollegen gemeinsam mit dem gewählten Mitarbeiter den Dienst teilen. Ausschlusskriterium ist hier K für Krankheit und U für Urlaub.

```
If Cells(ilauf, Tag).Value = dienstform And laufvariable <> ilauf Then
If mitdienst = “” Then mitdienst = “mit “
If mitdienst <> “mit “ Then mitdienst = mitdienst & “, “
mitdienst = mitdienst & Cells(ilauf, 1).Value
End If
```


Im ersten IF wird ein geprüft, ob die LAUFVARIABLE der internen Laufvariable ILAUF entspricht. Ist dies der Fall, so wird einfach weitergesprungen, da ansonsten der Mitarbeiter mit sich selbst als Kollegen Dienst hätte. Wurde bisher kein Eintrag als Kollege in die Variable MITDIENST abgelegt, so wird der Name einfach eingetragen, ansonsten wird der nächste Kollegenname mit einem Komma abgetrennt angehängt.

```
isprung:
Next ilauf
```

Hier endet die interne Laufschleife ; die Feststellung, welche Kollegen den Dienst teilen, ist abgeschlossen. Überschneidende Dienstformen werden in DPE3 nicht im Mitarbeiterplan abgebildet.

Nun beginnt ein umfangreicher Prüfungsbereich auf Zusatzdienste und Teamsitzungen. Zunächst wird abgeprüft, ob es sich um einen Zivildienstleistenden handelt, oder um eine Krankenpflegeschülerkraft, und ob diese zur Teamsitzung geladen werden oder nicht:

```
If Left$(Cells(laufvariable + 1, 1).Value, 3) = „KPS“ And Worksheets(“DEFINITIONEN”) _
.Range(“C101”) = 0 Then GoTo isprung2
If Left$(Cells(laufvariable + 1, 1).Value, 3) = “ZDL” And Worksheets(“DEFINITIONEN”) _
.Range(“C101”) = 0 Then GoTo isprung2

if mitdienst = “” Then
mitdienst = “Teamsitzung (“ & Worksheets(“DEFINITIONEN”).Range(“D97”).Value & “)”
If dienstform = “” Then
dienstdauer = Worksheets(“DEFINITIONEN”).Range(“D97”).Value

diensform = Worksheets(„DEFINITIONEN“).Range(„C96“).Value
End If , If von Dienstform beendet
Else
mitdienst = mitdienst & „ und Teamsitzung („ & _
Worksheets(„DEFINITIONEN“).Range(„D97“).Value & „)“
End If , If von Mitdienst beendet
End If , If von Teamsitzung beendet
```

Obiger Programmabschnitt erweitert den Übergabetext MITDIENST um den Teamsitzungseintrag.

```
isprung2:
Worksheets(„MIPLAN“).Cells(Tag, 3).Value = dienstform
If dienstzeit_anzeigen = 1 Then Worksheets(„MIPLAN“).Cells(Tag, 4).Value _
= dienstdauer

If bereitschaft = 1 Then
If zeile_zwei = bereitschaftssymbol Then
If mitdienst = „“ Then
mitdienst = „Bereitschaftsdienst“
Else
mitdienst = mitdienst & „ + Bereitschaft“
End If
End If
End If
Worksheets(„MIPLAN“).Cells(Tag, 5).Value = mitdienst
```

Neben dem Eintrag in den Mitarbeiterplan wird abgeprüft, ob in der zweiten Zeile der Planungstabelle für den Mitarbeiter die Buchstabenkombination für einen Bereitschaftsdienst eingegeben wurde. Der Bereitschaftsdienst ist vollkommen frei von jeder Berechnung im DPE3.

```
If dienstform = Worksheets(„DEFINITIONEN“).Range(„C107“).Value Or _
Cells(laufvariable, Tag + 62).Value <> „“ Then
If Len(Cells(laufvariable, Tag + 62).Value) >= 7 Then
For laeufer = 1 To Len(Cells(laufvariable, Tag + 62).Value)
untersuchung = Mid$(Cells(laufvariable, Tag + 62).Value, laeufer, 1)
```

```
If Asc(untersuchung) = 40 Then schnipp = Left$(Cells(laufvariable, Tag _
+ 62).Value, laeufer - 1)
If Asc(untersuchung) = 41 Then schnapp = _
Mid$(Cells(laufvariable, Tag + 62).Value, _
Len(schnipp) + 2, _
Len(Cells(laufvariable, Tag + 62).Value) - _
(Len(schnipp)) - 2)
Next laeufer
End If ‘ If von Besonderheiten Ende

If Len(schnapp) = 0 Then
mitdienst = mitdienst & Cells(laufvariable, Tag + 62).Value
Else
If Worksheets(„MIPLAN“).Cells(Tag, 4).Value = „“ Then _
Worksheets(„MIPLAN“).Cells(Tag, 4).Value = schnapp
If mitdienst = „“ Then
mitdienst = schnipp
Else
mitdienst = mitdienst & “ + “ & schnipp & “(“ & schnapp & “)”
End If
Worksheets(„MIPLAN“).Cells(Tag, 5).Value = mitdienst
End If , If von ...VALUE = „“ Ende
End If ‘ If von Len(schnapp) = 0 Ende

schnapp = “”
schnipp = “”
untersuchung = “”
mitdienst = “”
dienstdauer = “”
```

Obige Programmzeilen prüfen im versteckten Bereich der Excel-Tabelle, ob für den jeweiligen Dienst noch weitere Einträge vorgenommen wurden, in Form von Sonderdiensten oder frei definierten Dienstformen. Der Aufbau rund um SCHNIPP und SCHNAPP sorgt dafür, dass die Uhrzeitwerte aus den Klammern ausgeschnitten werden, um im Klartext im Mitarbeiterplan angedruckt werden zu können. Die Variablen werden abschließend noch zurückgesetzt.

```
If Range(„AI1“).Value = „\“ Then Range(„AI1“).Value = „/“ Else Range(„AI1“).Value = „\“
Next Tag
```

Der Range-Befehl produziert einen abwechselnden \ / Balken auf der Planungstabelle und signalisiert so etwas wie ein Lebenszeichen, da dieser Vorgang einige Zeit in Anspruch nimmt und Microsoft Excel das System zuweilen stark blockiert. Die Tages-Schleife endet hier.

```
If Worksheets(„DEFINITIONEN“).Range(„C104“).Value = 1 Then Worksheets(“MIPLAN”).P
rintOut

sprungmarke:
Next laufvariable

Range(“AI1”).Value = “”
Application.Cursor = xlDefault
Application.ScreenUpdating = True
Application.Calculate
Range(“D9”).Select
```

Der letzte Befehl vor der SPRUNGMARKE sorgt für einen Ausdruck auf dem Windows-Standard-Drucker, sofern in der DEFINTIONS-Tabelle der Wert auf 1 gesetzt wurde. Die Hauptschleife endet hier. Nach erfolgtem, komplettem Ausdruck wird die Zelle AI1 mit dem \ beziehungsweise / Balken zurückgesetzt und der Mauszeiger wieder von der Sanduhr auf Standard umgestellt. Die Bildschirmdarstellung wird wieder eingeschaltet und eine Neuberechnung durchgeführt, für den Fall, dass sich Daten innerhalb des Excel-Blatts verändert haben.

Sicht vereinfachen
Die Schaltfläche mit dem Symbol eines Aktenschranks blendet die dritte Zeile in der Planungstabelle aus und macht den Plan so übersichtlicher.

```
Sub sicht_vereinfachen()
If Worksheets(„PLAN“).Rows(„1:3“).EntireRow.Hidden = True Then
Call Einblendung
Else
Call Ausblendung
End If
End Sub
```

12.3 Modul 2

Im Modul 2 findet sich der Commissaire aux comptes, zu Deutsch der Rechnungsprüfer. Der Dienstplan wird auf die Einhaltung der Sollstellung, der Mitarbeiterwünsche und Überstunden in Abhängigkeit der vergebenen Urlaubstage qualitativ analysiert. Für künftige Versionen ist hier sicherlich noch viel Spielraum für Verbesserung – beispielsweise der Ausgabe von Vorschlägen für ein „Dienst-Balancing“.

Das CAC-Modul gibt den Report nicht direkt in Microsoft Excel aus, sondern überträgt die Informationen in eine Text-Datei und öffnet diese nach Abschluss der Erhebung mit einem Texteditor. Da die Geschwindigkeitsprobleme, die zwischenzeitlich auftraten, gelöst sind, lässt sich auch diese Funktion problemlos direkt mit Excel abbilden.

mitarbeiter(22)	STRING	Namen aller Mitarbeiter
soll(22)	DOUBLE	Die Solldienstzeiten aller 22 Mitarbeiter
ist(22)	DOUBLE	Die tatsächlichen Monatsstunden aller 22 Mitarbeiter
wunschwert(22)	INTEGER	Wünsche der Mitarbeiter
zaehler	BYTE	Laufvariable 1 für Schleifen < 255
laufvariable	BYTE	Laufvariable 2 für Schleifen < 255
antwort	BYTE	Antwortwert der MSGBOX
aussagetext(22)	STRING	Ergebnisse für jeden Mitarbeiter
regel(5)	INTEGER	Zahlenwerte der Mitarbeiterwünsche
akt_dienst	STRING	Aktuell ausgelesener Dienst aus dem Plan
schnittlaenge	BYTE	Länge der Dienstbezeichnungen
u	STRING	zu übergebender Tag
dienst(7)	STRING	die sieben Dienstformen des DPE3
fehldienst	STRING	Welche Dienstarten werden als fehlend gefunden
alle_u	INTEGER	Anzahl gefundener Urlaubstage im Plan
pw_wunsch	BYTE	Punktwert aus DEFINITIONEN für Erfüllung eines Wunsches
pw_gegen_aus	BYTE	Punktwert aus DEFINITIONEN bei Dienst gegen ausdr. Wunsch
pw_bei_ungern	BYTE	Punktwert aus DEFINITIONEN bei Dienst entgegen Wunsch
pw_bei_kein	BYTE	Punktwert aus DEFINITIONEN bei Dienst an einem „ganz frei“ Tag
notw_a ... _w	INTEGER	Anzahl benötigter Dienste für diesen Tag
Tag	INTEGER	Schleifenvariable für Tage
akt_tag	VARIANT	Ausgelesener Wert der aktuellen Tag
wunschdienst	VARIANT	Gewünschte Dienstform
regel_zaehler	BYTE	Kleine Schleifenvariable für die fünf Wünsche
d	INTEGER	Dateinummer für die Ausgabe
a	INTEGER	Laufvariable 3 für eine weitere Schleife
ergebnis	INTEGER	Das Summenergebnis der CAC-Prüfung

```
,   *** Sicherheitsabfrage
antwort = MsgBox(„Soll eine Plananalyse durchgeführt werden ?“, vbYesNo, „Frage!“)
If antwort = 7 Then Exit Sub
```

Bevor mit dem Programmablauf begonnen wird, muss der Benutzer zunächst mit JA dies ausdrücklich bestätigen.

```
,   Aufbau der Matrix für Dienstnamen
schnittlaenge = Worksheets(„DEFINITIONEN“).Cells(91, 3).Value
For laufvariable = 1 To 7
dienst(laufvariable) = Worksheets(„DEFINITIONEN“).Cells(17 + laufvariable, 3).Value
Next laufvariable

‘   Einlesen von Stammwerten
pw_wunsch = Worksheets(„DEFINITIONEN“).Range(„C139“).Value
pw_gegen_aus = Worksheets(„DEFINITIONEN“).Range(„C140“).Value
pw_bei_ungern = Worksheets(„DEFINITIONEN“).Range(„C141“).Value
pw_bei_kein = Worksheets(„DEFINITIONEN“).Range(„C142“).Value
```

Die benötigten Werte werden ausgelesen. Im obigen Abschnitt zunächst, wie die Dienste überhaupt bezeichnet sind. Anschließend die Punktwerte für die CAC-Analyse.

```
,   Prüfung ob Dienste korrekt besetzt sind
For laufvariable = 66 To 96
notw _ a  = notw_a + Cells(96,   laufvariable).Value - Cells(75, laufvariable - 62).Value
notw _ b  = notw_b + Cells(99,   laufvariable).Value - Cells(78, laufvariable - 62).Value
notw _ c  = notw_c + Cells(97,   laufvariable).Value - Cells(76, laufvariable - 62).Value
notw _ d  = notw_d + Cells(101, laufvariable).Value - Cells(80, laufvariable - 62).Value
notw _ e  = notw_e + Cells(100, laufvariable).Value - Cells(79, laufvariable - 62).Value
notw _ w  = notw_w + Cells(98,   laufvariable).Value - Cells(77, laufvariable - 62).Value
Next laufvariable
```

Im „versteckten“ Bereich der Excel-Tabelle werden die effektiven Sollstellungen ausgelesen und für jeden Tag mit der realen Besetzung überprüft. Die Werte werden in den notw_<Dienstname> Variablen zwischengespeichert.

```
,   Hauptschleife initiieren
For laufvariable = 9 To 72 Step 3

zaehler = zaehler + 1
mitarbeiter(zaehler) = Cells(laufvariable, 1).Value
soll(zaehler) = CDbl(Cells(laufvariable, 3).Value)
ist(zaehler) = CDbl(Cells(laufvariable, 37).Value)
If mitarbeiter(zaehler) = “Leer” Or mitarbeiter(zaehler) = “” Then GoTo sprungmarke1
regel(1) = CInt(Left$  (Cells(laufvariable, 49).Value, 3))
regel(5) = CInt(Right$(Cells(laufvariable, 49).Value, 3))
regel(2) = CInt(Mid$   (Cells(laufvariable, 49).Value, 5, 3))
regel(3) = CInt(Mid$   (Cells(laufvariable, 49).Value, 9, 3))
regel(4) = CInt(Mid$   (Cells(laufvariable, 49).Value, 13, 3))
```

Für jeden Mitarbeiter müssen zunächst die Stammwerte eingelesen werden. Hinter der Feldvariable REGEL(n) verbergen sich die fünf Wünsche, die zu jedem Mitarbeiter eingetragen werden können. Mit dem Befehl CDbl(<ZELLE>) wird dafür gesorgt, dass der Datums- und Zeitwert von Excel korrekt in VBA übernommen wird. Namenlose Mitarbeiter werden vom Programm ignoriert.

```
,   Analyse der Wunschparameter
For Tag = 4 To 35
‘   Den Wochentagwert des aktuellen Datums auslesen
akt_tag = Weekday(Cells(5, Tag).Value)
```

Die interne Schleife „Tag“ prüft nun jeden einzelnen Tag des Plans ab. Die Variable akt_tag hält den Tageswert des aktuellen Datums.

```
,  Ist der Tag ein Feiertag, so gilt diese Einstellung
If Cells(7, Tag).Value = „ft“ Then akt_tag = 8
akt_dienst = Left$(Cells(laufvariable, Tag).Value, schnittlaenge)
‘   Urlaubstage auf dem Plan zählen
```



```
If akt_dienst = Worksheets(„DEFINITIONEN“).Range(„C24“).Value Then
alle_u = alle_u + 1
GoTo sprungmarke2
End If
```

Neben der Feststellung, ob ein Feiertag vorliegen, wird über die Variable alle_u die gesamten Urlaubstage, die im Plan gefunden werden aufaddiert. Der Sprungbefehl zur sprungmarke2 unterbindet eine Wunsch-Prüfung, da Urlaubstage nicht als Wunschdienst zugelassen sind.

```
For regel_zaehler = 1 To 5

If regel(regel_zaehler) < 100 Then GoTo sprungmarke1

,  Wunsch für Sonntag entdeckt
If regel(regel_zaehler) >= 100 And regel(regel_zaehler) < 200 And akt_tag = 1 Then
If Mid$(CStr(regel(regel_zaehler)), 2, 1) = „0“ And akt_dienst <> „“ Then
   GoSub reg_null
   End If
If Mid$(CStr(regel(regel_zaehler)), 2, 1) = “1” And akt_dienst <> “” Then
   GoSub reg_eins
   End If
If Mid$(CStr(regel(regel_zaehler)), 2, 1) = “2” And akt_dienst <> “” Then
   wunschwert(zaehler) = wunschwert(zaehler) + 1
   End If
If Mid$(CStr(regel(regel_zaehler)), 2, 1) = “3” And akt_dienst <> “” Then
   wunschwert(zaehler) = wunschwert(zaehler) + 2
   End If
‘  Test auf bevorzugter Dienst
If Mid$(CStr(regel(regel_zaehler)), 2, 1) = “5” And akt_dienst <> “” Then
   GoSub reg_fuenf
   End If
‘  Test auf “ungern” Dienst
If Mid$(CStr(regel(regel_zaehler)), 2, 1) = “4” And akt_dienst <> “” Then
   GoSub reg_vier
   End If
GoTo sprungmarke1
End If


‘  Wunsch für Montag entdeckt
If regel(regel_zaehler) >= 200 And regel(regel_zaehler) < 300 And akt_tag = 2 Then
If Mid$(CStr(regel(regel_zaehler)), 2, 1) = “0” And akt_dienst <> “” Then
   GoSub reg_null

   End If
If Mid$(CStr(regel(regel_zaehler)), 2, 1) = “1” And akt_dienst <> “” Then
   GoSub reg_eins
   End If
If Mid$(CStr(regel(regel_zaehler)), 2, 1) = “2” And akt_dienst <> “” Then
   wunschwert(zaehler) = wunschwert(zaehler) + 1
   End If
If Mid$(CStr(regel(regel_zaehler)), 2, 1) = “3” And akt_dienst <> “” Then
   wunschwert(zaehler) = wunschwert(zaehler) + 2
   End If
‘ Test auf bevorzugter Dienst
If  Mid$(CStr(regel(regel_zaehler)), 2, 1) = “5” And akt_dienst <> “” Then
    GoSub reg_fuenf
    End If
‘  Test auf “ungern” Dienst
If Mid$(CStr(regel(regel_zaehler)), 2, 1) = “4” And akt_dienst <> “” Then
```

```
   GoSub reg_vier
   End If
GoTo sprungmarke1
End If

{...} für weitere Tage entsprechend

‘  Wunsch für JEDEN TAG entdeckt. Wünsche wie ‘Jeden Tag keinen Dienst’ werden
‘  vom Programm genußvoll ignoriert. Nur bei Angabe einer Dienstform ist dieser
‘  Eintrag zulässig.
If regel(regel_zaehler) >= 940 Then
‘  Test auf bevorzugter Dienst
If Mid$(CStr(regel(regel_zaehler)), 2, 1) = “5” And akt_dienst <> “” Then
    GoSub reg_fuenf
    End If
‘ Test auf “ungern” Dienst
If Mid$(CStr(regel(regel_zaehler)), 2, 1) = “4” And akt_dienst <> “” Then
    GoSub reg_vier
    End If
End If

sprungmarke1:
Next regel_zaehler
```

Die gesamte Regelanalyse, die viele Zeilen des Quellcodes ausmachen, macht nichts weiter als zu prüfen, ob für den aktuellen Tag ein Wunsch vorliegt. Im Abschnitt 10.6.4 findet sich die komplette Verschlüsselung, die für den Aufbau der Mitarbeiterwünsche verwendet wird.

```
sprungmarke2:
Next Tag
Next laufvariable
```

Sprungmarke mit Beendigung der Schleifen für den jeden einzelnen Tag und jeden einzelnen Mitarbeiter. Der Programmablauf wird somit maximal 22x31 = 682-mal wiederholt. Im Programmverlauf werden wiederholt Unterprogramme aufgerufen, die für den Aufbau der Mitarbeitergebnisse verwendet werden.

```
,  Ausgabe
d = FreeFile
Open Worksheets(„DEFINITIONEN“).Cells(146, 3).Value For Output As #d
Print #d, “Plananalyse vom      : “ & Date$ & “ “ & Time$
```

Mit dem Befehl Open <Dateiname> For Output as <Dateinummer> wird die Ausgabedatei erzeugt. Ist diese bereits vorhanden, so wird sie ohne Dialogfeld überschrieben und durch die neue Analyse ersetzt.

```
For a = 1 To 22
‘  Wurde die Sollarbeitszeit über die einstellbaren Parameter hinweg überzogen
If soll(a) > (ist(a) + (ist(a) * Worksheets(„DEFINITIONEN“).Cells(92, 3).Value) _
/ 100) Then ergebnis = ergebnis - 1
ergebnis = ergebnis + wunschwert(a)
‘  Wurden alle Schichten besetzt ?
Next a
```

Die Parameter aus der DEFINITIONEN-Tabelle, die für den Farbwechsel bei Unter- beziehungsweise Überplanung des Stundenwerts eines einzelnen Mitarbeiters verantwortlich ist, werden auch hier für eine Punktwertberechnung verwendet. Für jede Über- beziehungsweise Unterplanung außerhalb des zulässigen Prozentbereichs wird ein Punkt in Abzug gebracht.

```
If notw_a > 0 Then fehldienst = fehldienst & Worksheets(„DEFINITIONEN“).Cells(18, 3).Value & „ „
If notw_b > 0 Then fehldienst = fehldienst & Worksheets(„DEFINITIONEN“).Cells(19, 3).Value & „ „
If notw_c > 0 Then fehldienst = fehldienst & Worksheets(„DEFINITIONEN“).Cells(20, 3).Value & „ „
If notw_d > 0 Then fehldienst = fehldienst & Worksheets(„DEFINITIONEN“).Cells(21, 3).Value & „ „
If notw_e > 0 Then fehldienst = fehldienst & Worksheets(„DEFINITIONEN“).Cells(22, 3).Value & „ „
```



```
If notw_w > 0 Then fehldienst = fehldienst & Worksheets(“DEFINITIONEN”).Cells(23, 3).Value & “ “
ergebnis = ergebnis - (notw_a + notw_b + notw_c + notw_d + notw_e + notw_w)
```

Für jeden Dienst, der nicht entsprechend der Sollbesetzung durch den Dienstplaner umgesetzt wurde, wird ebenfalls ein Punkt in Abzug gebracht. Gleichzeitig wird in der Zeichenkette „fehldienst“ die fehlende Dienstform benannt.

```
If alle_u > Worksheets(„DEFINITIONEN“).Range(„C143“).Value Then
ergebnis = ergebnis + (alle_u - Worksheets(“DEFINITIONEN”).Range(“C143”).Value)
End If
```

Um bei der qualitativen Analyse Engpässe durch Urlaubsperioden auszugleichen, lässt sich in der Tabelle DEFINITIONEN ein Wert festlegen, ab wie viel gefundenen Urlaubstagen jeder einzelne Urlaubstag überhab dieses Schwellwert als Pluspunkt für die Analyse aufaddiert werden kann.

```
If Worksheets(„DEFINITIONEN“).Range(„C144“).Value = 1 Then _
Print #d, “Erzielter Punktwert : “ & CStr(ergebnis) & vbCr
Print #d, “Fehlende Dienste in : “ & fehldienst
Print #d, “Anzahl Urlaubstage..: “ & CStr(alle_u)
Print #d, „-----------------------------------------------------------“
If Worksheets(“DEFINITIONEN”).Range(“C138”).Value = 1 Then
For a = 1 To 22
Print #d, “Mitarbeiter.........: “ & mitarbeiter(a)
Print #d, aussagetext(a)
Next a
End If
Print #d, ergebnistext
‘  Die Datei muss wieder geschlossen werden
Print #d, “”
Print #d, “Erstellt mit dem ‘Dienstplan in Excel’”
Close
```

Kopf der Analyse in die Datei ausgeben. Der Umfang der CAC-Analyse wird über Parameter in der Tabelle DEFINITONEN gesteuert.

```
,  Ergebnis der Prüfung im Texteditor öffnen
Shell (Worksheets(„DEFINITIONEN“).Cells(145, 3).Value & „ „ & _
Worksheets(„DEFINITIONEN“).Cells(146, 3).Value), vbNormalFocus
Exit Sub
```

Ergebnisdatei mit einem Texteditor öffnen.

```
,  Unterprogramme
reg_null:
wunschwert(zaehler) = wunschwert(zaehler) - pw_bei_kein
GoSub wasfuereintag
aussagetext(zaehler) = aussagetext(zaehler) & „Dienst am „ & u & „, obwohl kein „ & _
„  Dienst gewünscht wurde am „ & CStr(Cells(5, Tag).Value) & „ .“ & vbCrLf

Return
reg_eins:
wunschwert(zaehler) = wunschwert(zaehler) - pw_bei_ungern
GoSub wasfuereintag
aussagetext(zaehler) = aussagetext(zaehler) & “Dienst am “ & u & “, obwohl für “ & _
“  diesen Tag als ungern deklariert am “ & CStr(Cells(5, Tag).Value) & “.” & vbCrLf
Return
reg_vier:
‘  Feststellung welchen Dienst hätte der Mitarbeiter den ungern
wunschdienst = CInt(Right$(regel(regel_zaehler), 1))
For a = 1 To 7
If dienst(a) = akt_dienst And a = wunschdienst Then
```

```
wunschwert(zaehler) = wunschwert(zaehler) - pw_gegen_aus
aussagetext(zaehler) = aussagetext(zaehler) & “Ungewünschte Dienstform am “
& CStr(Cells(5, Tag).Value) & vbCrLf
Exit For
End If
Next a
Return

reg_fuenf:
‘  Feststellung welchen Dienst hätte der Mitarbeiter den gern
wunschdienst = CInt(Right$(regel(regel_zaehler), 1))
For a = 1 To 7
If dienst(a) = akt_dienst And a = wunschdienst Then
wunschwert(zaehler) = wunschwert(zaehler) + pw_wunsch
Exit For
End If
Next a
Return

wasfuereintag:
Select Case akt_tag
Case 1
u = “Sonntag”
Case 2
u = „Montag“
Case 3
u = “Dienstag”
Case 4
u = “Mittwoch”
Case 5
u = „Donnerstag“
Case 6
u = “Freitag”
Case 7
u = „Samstag“
Case 8
u = “Feiertag”
Case 9
u = “Täglich”
End Select
Return

Exit Sub
```

Die Unterprogramme steuern den Aufbau der Ergebnistexte für jeden Mitarbeiter. Unter der Konstante vbCrLf verbirgt sich ein quasi drücken der Eingabetaste – „Carriage return Line feed“.

```
fehler:
If Err.Number = 76 Then
MsgBox “Fehler beim Zugriff auf den Texteditor. Wahrscheinlich ist der Pfad in der “ _
& “DEFINITIONS-Tabelle falsch gewählt. Windows NT und 2000 haben als Standardpfad “_
& “\WINNT, während Windows ´9x und XP als Pfad \WINDOWS verwenden.”, vbCritical
Exit Sub
End If
MsgBox “Ein Fehler ist aufgetreten : “ & Err.Number, vbCritical
End Sub
```

Ein kleiner Fehlerhandler, der den wahrscheinlichsten Fehler abfangen soll: Der Texteditor konnte nicht gefunden werden.

12.4 Modul 3

Im Modul 3 sind alle Programmteile zusammengefasst, die für Steuerung der Dialogfelder verantwortlich sind. Ein glücklicher Umstand von Microsoft Excel macht das Auslesen von Werten nach dem Schließen des Fensters einfach: die Werte, die im Dialogfeld eingetragen, beziehungsweise ausgewählt wurden, werden nicht entfernt. Würde im VBA- Quelltext nicht laufend eine „Bereinigung“ des Fensterinhalts durchgeführt werden, so würden selbst nach einem Schließen der Excel-Datei die Werte in den Dialogfeldern erhalten bleiben.

Planungshelfer

regel(5)	INTEGER	Inhalt der fünf Mitarbeiterwünsche
akt_row	INTEGER	Reihenwert der aktuellen Cursorposition
akt_col	INTEGER	Spaltenwert der aktuellen Cursorposition
a	BYTE	Laufvariable 1 (< 255)
b	BYTE	Wurde eine Regel für diesen Tag gefunden (1=JA,0=NEIN)
akt_tag	INTEGER	Wochentagwert des aktuellen Tages
regel_zaehler	BYTE	Laufvariable 2 (< 255)
aussage	STRING	Ergebnissatz „Aus den Wünschen“
dienst(7)	STRING	Feldvariable mit den Dienstformnamen

```
Sub Dialog2_Dialogfeldrahmen1_BeiAnzeigen()

DialogSheets(„PLANUNGSHELFER“).[Titel].Caption = „“
DialogSheets(„PLANUNGSHELFER“).[Zeitwert].Caption = „00:00“
DialogSheets(„PLANUNGSHELFER“).[Standarddienst].Value = 8
DialogSheets(„PLANUNGSHELFER“).[Sonderdienstform].Value = 10
DialogSheets(„PLANUNGSHELFER“).[Malparameter].Value = 1
DialogSheets(„PLANUNGSHELFER“).[Wunsch].Caption = „“
DialogSheets(„PLANUNGSHELFER“).[HAPPY].Visible = False
DialogSheets(„PLANUNGSHELFER“).[OK].Visible = False
DialogSheets(„PLANUNGSHELFER“).[SAD].Visible = False
DialogSheets(„PLANUNGSHELFER“).[NO].Visible = False
DialogSheets(„PLANUNGSHELFER“).[HERZ].Visible = False
DialogSheets(„PLANUNGSHELFER“).[HAND].Visible = False
```

Alte Fensterdaten werden zurückgesetzt, grafische Symbole ausgeblendet. Die Anzeige für den Standarddienst wird mit dem Wert 8 auf „[keinen Dienst]“ gesetzt, ebenso wird mit dem Wert 10 der Sonderdienstform „[keinen Dienst]“ ausgewählt.

```
akt_row = ActiveCell.Row
akt_col = ActiveCell.Column
```

Auslesen der aktuellen Cursorposition.

```
, Wunschwert auslesen
regel(1) = CInt(Left$(Cells(akt_row, 49).Value, 3))
regel(5) = CInt(Right$(Cells(akt_row, 49).Value, 3))
regel(2) = CInt(Mid$(Cells(akt_row, 49).Value, 5, 3))
regel(3) = CInt(Mid$(Cells(akt_row, 49).Value, 9, 3))
regel(4) = CInt(Mid$(Cells(akt_row, 49).Value, 13, 3))
```

Nachdem die aktuelle Reihe bekannt ist, können die Wunschwerte für den Mitarbeiter ausgelesen werden. Aus dem String der Wünsche werden die durch Komma geteilten Werte ausgelesen und in Standard-integer Zahlen verwandelt.

```
For a = 1 To 5
If regel(a) <> 0 Then b = 1
Next a
If b = 0 Then Exit Sub
```

Wird kein Wunsch für den Mitarbeiter gefunden, so kann der Aufbau der Wunschansicht gleich beendet werden. Das Auslesen der gewählten Werte aus dem Fenster geschieht erst durch ein Klick auf „OK“ im Dialogfenster.

```
akt_tag = Weekday(Cells(5, akt_col).Value)
If Cells(7, akt_col).Value = „ft“ Then akt_tag = 8
```

Feststellung welcher Wochentag für den aktuellen Tag vorliegt. Handelt es sich um einen Feiertag so wird der Wert 8 verwendet. Die folgende Prüfung auf das Vorhandensein der Mitarbeiterwünsche entspricht dem Quelltext aus der qualitativen Dienstplanprüfung (CAC)!

```
{...} If regel(regel_zaehler) < 100 Then GoTo sprungmarke1 {...}
```

```
, Ausgabe vorbereiten | Programmablauf hier abbrechen
DialogSheets(„PLANUNGSHELFER“).[Wunsch].Caption = aussage
Exit Sub
```

In AUSSAGE werden die Wünsche des Mitarbeiters für diesen Tag im Klartext abgelegt und im Dialogfeld unter „Aus den Wünschen“ angezeigt. Zusätzlich zur Textausgabe werden Symbole im Dialogfeld angezeigt, die grafisch die Wünsche präsentieren. Je nachdem, welche Wünsche gefunden wurden, wird hier in ein Unterprogramm verwiesen, welches die Bitmap-Grafik mit dem Symbol einblendet.

```
analyser:
If Mid$(CStr(regel(regel_zaehler)), 2, 1) = „0“ Then GoSub keinen_dienst
If Mid$(CStr(regel(regel_zaehler)), 2, 1) = „1“ Then GoSub ungern_dienst
If Mid$(CStr(regel(regel_zaehler)), 2, 1) = „2“ Then GoSub gern_dienst
If Mid$(CStr(regel(regel_zaehler)), 2, 1) = „3“ Then GoSub sehr_gern_dienst
If Mid$(CStr(regel(regel_zaehler)), 2, 1) = „4“ Then GoSub ungern_dienstform
If Mid$(CStr(regel(regel_zaehler)), 2, 1) = „5“ Then GoSub gern_dienstform
Return

keinen_dienst:
aussage = aussage & „ Keinen Dienst.“
DialogSheets(„PLANUNGSHELFER“).[NO].Visible = True
Return

ungern_dienst:
aussage = aussage & “ Ungern Dienst.”
DialogSheets(„PLANUNGSHELFER“).[SAD].Visible = True
Return

gern_dienst:
aussage = aussage & “ Gern Dienst.”
DialogSheets(„PLANUNGSHELFER“).[OK].Visible = True
Return

sehr_gern_dienst:
aussage = aussage & “ Sehr gern Dienst.”
DialogSheets(„PLANUNGSHELFER“).[HAPPY].Visible = True
Return

ungern_dienstform:
aussage = aussage & “ Ungern : “ & _
dienst(CInt(Right$(regel(regel_zaehler), 1)))
DialogSheets(“PLANUNGSHELFER”).[HAND].Visible = True
Return

gern_dienstform:
aussage = aussage & “ Gern : “ & _
dienst(CInt(Right$(regel(regel_zaehler), 1)))
DialogSheets(“PLANUNGSHELFER”).[HERZ].Visible = True
Return

End Sub
```


Mitarbeiter-Dialogfeld

```
Sub MITARBEITER_Dialogfeldrahmen1_BeiAnzeigen()
On Error Resume Next
DialogSheets(“MITARBEITER”).[mitarbeitername].Caption = v_mitarbeitername
DialogSheets(“MITARBEITER”).[kartennummer].Caption = v_kartennummer
DialogSheets(“MITARBEITER”).[dienstform].Caption = v_mita_an
DialogSheets(“MITARBEITER”).[NWACHE].Caption = v_nachtwache
If v_mita_az = “1/1” Then DialogSheets(“MITARBEITER”).[DF1].Value = “1”
If v_mita_az = “3/4” Then DialogSheets(“MITARBEITER”).[DF2].Value = “1”
If v_mita_az = “1/2” Then DialogSheets(“MITARBEITER”).[DF3].Value = “1”
If v_mita_az = “1/4” Then DialogSheets(“MITARBEITER”).[DF4].Value = “1”
If v_mita_az = „ WT“ Then DialogSheets(„MITARBEITER“).[DF5].Value = „1“
End Sub
```

Das Mitarbeiterdialogfeld wird mit einem Doppelklick auf den Namen des Mitarbeiters aus dem Planungsteil aufgerufen. Alle in diesem Programmabschnitt verwendeten Variablen sind als Globalvariablen deklariert und stehen somit dem ganzen Projekt zur Verfügung. Da nur eine Dienstzeit (1/1 bis 1/4) vorliegen kann, ist es ausreichend nur den einen Wert der Radio-Buttons auf 1 zu setzen. Da stets nur ein Radio-Button innerhalb eines Rahmens aktiv sein kann, ist die Zuweisung der Nullwerte an dieser Stelle überflüssig.

Mitarbeiter-Dialogfeld mit OK

```
Sub MITARBEITER_Schaltfläche2_BeiKlick()
If DialogSheets(„MITARBEITER“).[mitarbeitername].Caption = „“ Then
v_mitarbeitername = „Leer“
Else
v_mitarbeitername = DialogSheets(„MITARBEITER“).[mitarbeitername].Caption
End If
If DialogSheets(„MITARBEITER“).[kartennummer].Caption = „“ Then
v_kartennummer = „Def“
Else
v_kartennummer = DialogSheets(„MITARBEITER“).[kartennummer].Caption
End If
v_mita_an = DialogSheets(„MITARBEITER“).[dienstform].Caption
v_nachtwache = DialogSheets(„MITARBEITER“).[NWACHE].Caption
If DialogSheets(„MITARBEITER“).[DF1].Value = „1“ Then v_mita_az = „1/1“
If DialogSheets(„MITARBEITER“).[DF2].Value = „1“ Then v_mita_az = „3/4“
If DialogSheets(„MITARBEITER“).[DF3].Value = „1“ Then v_mita_az = „1/2“
If DialogSheets(„MITARBEITER“).[DF4].Value = „1“ Then v_mita_az = „1/4“
If DialogSheets(„MITARBEITER“).[DF5].Value = „1“ Then v_mita_az = „ WT“
End Sub
```

Wird das Dialogfeld mit „OK“ geschlossen so werden die aktuellen Fensterwerte ausgelesen und in den Globalvariablen gespeichert. Der Programmverlauf wird nach dem Klick auf „OK“ im aufrufenden Programmabschnitt fortgeführt.

Mitarbeiter Löschen

```
Sub MITARBEITER_Schaltfläche3_BeiKlick()
v_mitarbeitername = „Leer“
v_kartennummer = „Stnr“
v_mita_an = „!“
v_mita_az = „!“
End Sub
```

Wurde die Schaltfläche „Löschen“ im Mitarbeiter-Dialogfeld gedrückt, so werden alle Einträge auf den Standardwert zurückgesetzt.

Ausgewählte Dienstformen

```
Sub Standarddienform_Aenderung()
dienstart = 1
End Sub

Sub Sonderdienstform_Aenderung()
dienstart = 2
End Sub

Sub Frei_Definition_Titel_Aenderung()
dienstart = 3
End Sub

Sub Zeitwert_Aenderung()

dienstart = 3
End Sub

Sub Keinen_Dienst_BeiKlick()
dienstart = 4
Call Planungshelfer_OK
End Sub
```

Je nachdem welches Segment im Planungshelfer zuletzt ausgewählt wurde, wird hier entsprechend der Globalvariablen „dienstart" mit einem Wert versorgt. So lässt sich im aufrufenden Programmabschnitt feststellen, welche Dienstform der Benutzer nun tatsächlich ausgewählt hat. Wird die Schaltfläche „Keinen Dienst" ausgewählt, so wird neben der Wertzuweisung automatisch der Programmcode für die geklickte „OK"-Schaltfläche aufgerufen, die das Fenster schließt.

Planungshelfer OK

```
Sub Planungshelfer_OK()
If dienstart = 3 Then
If Len(DialogSheets(„PLANUNGSHELFER").[Zeitwert].Caption) <> 5 Then
Beep
DialogSheets(„PLANUNGSHELFER").[Zeitwert].Caption = „00:00"
Exit Sub
End If
If Mid$(DialogSheets("PLANUNGSHELFER").[Zeitwert].Caption, 3, 1) <> ":" Then
Beep
DialogSheets("PLANUNGSHELFER").[Zeitwert].Caption = "00:00"
Exit Sub
End If
If DialogSheets(„PLANUNGSHELFER").[Titel].Caption = „" Then
Beep
Exit Sub
End If
End If
ActiveWorkbook.DialogSheets(„PLANUNGSHELFER").Hide
End Sub
```

Wird die Schaltfläche „OK" im Planungshelfer geklickt, so wird eine einfache Prüfung der eingegeben Werte aufgrund der Länge der Zeitwerte vorgenommen. Wird ein Eintrag mit ungültiger Zeichenanzahl gefunden, so wird der Standard-Warnton mit dem Befehl BEEP ausgegeben und statt dem ungültigen Wert der Eintrag „00:00" verwendet. Eine weiterreichende Prüfung der Eingaben, beispielsweise auf einen Doppelpunkt in der Mitte oder das der Minutenwert nur bis 60 zulässig ist, entfällt.

Monatsauswahl

```
Sub Schaltfläche2_BeiKlick()
gewaehlter_monat = DialogSheets(„Dialog1").[monatswahl].Value
ActiveWorkbook.DialogSheets(„Dialog1").Hide
End Sub
```

Das kleine Fenster mit der Auswahl des Monats für den neuen Plan wird mit OK geschlossen und der Wert in der Globalvariable gewaehler_monat abgespeichert.

Mitarbeiterwünsche - I

```
Sub wunschplan_aktivieren()
ActiveWorkbook.DialogSheets(„MITARBEITER").Hide
ActiveWorkbook.DialogSheets(„WASDARFSSEIN").Show
End Sub
```

Das Dialogfeld Mitarbeiter wird versteckt und das Fenster zur Eingabe der Mitarbeiterwünsche in den Vordergrund gebracht.

Mitarbeiterwünsche – II

```
Sub WASDARFSSEIN_Dialogfeldrahmen1_BeiAnzeigen()
Dim akt_row As Byte
Dim regel(7) As Integer
Dim laufvariable As Integer
Dim wache As String
akt_row = ActiveCell.Row

'  Regelwerk für den betreffenden Mitarbeiter aufbauen
regel(1) = CInt(Left$(Cells(akt_row, 49).Value, 3))
regel(5) = CInt(Right$(Cells(akt_row, 49).Value, 3))
regel(2) = CInt(Mid$(Cells(akt_row, 49).Value, 5, 3))
regel(3) = CInt(Mid$(Cells(akt_row, 49).Value, 9, 3))
regel(4) = CInt(Mid$(Cells(akt_row, 49).Value, 13, 3))
,  Block

If regel(1) = 0 Then
DialogSheets("WASDARFSSEIN").[D1].Value = 0
DialogSheets("WASDARFSSEIN").[T1].Value = 0
DialogSheets("WASDARFSSEIN").[R1].Value = 0
DialogSheets("WASDARFSSEIN").[Dienst1].Value = 0
Else
DialogSheets("WASDARFSSEIN").[D1].Value = CInt(Mid$(CStr(regel(1)), 2, 1)) + 1
DialogSheets("WASDARFSSEIN").[T1].Value = CInt(Left$(CStr(regel(1)), 1))
DialogSheets("WASDARFSSEIN").[R1].Value = 1
DialogSheets("WASDARFSSEIN").[Dienst1].Value = CInt(Right$(CStr(regel(1)), 1))
End If

{...}

End Sub
```

Wird das Fenster für die Eingabe der Mitarbeiterwünsche geöffnet, so müssen zunächst die bereits in der Tabelle hinterlegten Wünsche ausgelesen werden und hier in die grafische Ansicht gebracht werden.

Mitarbeiterwünsche – III

```
Sub WASDARFSSEIN_Schaltfläche2_BeiKlick()

Dim akt_row As Byte
Dim laufvariable As Integer
Dim regel As String
akt_row = ActiveCell.Row
If DialogSheets(„WASDARFSSEIN“).[R1].Value = 1 Then
regel = CStr(DialogSheets(“WASDARFSSEIN”).[T1].Value)
regel = regel & CStr(DialogSheets(“WASDARFSSEIN”).[D1].Value - 1)
If DialogSheets(„WASDARFSSEIN“).[D1].Value = 5 Or _
   DialogSheets(„WASDARFSSEIN“).[D1].Value = 6 Then
   regel = regel & CStr(DialogSheets(“WASDARFSSEIN”).[Dienst1].Value) & “,”
   Else
   regel = regel & “0,”
   End If
Else
regel = „000,“
End If

{...}

If DialogSheets(„WASDARFSSEIN“).[R5].Value = 1 Then
regel = regel & CStr(DialogSheets(“WASDARFSSEIN”).[T5].Value)
regel = regel & CStr(DialogSheets(“WASDARFSSEIN”).[D5].Value - 1)
If DialogSheets(„WASDARFSSEIN“).[D5].Value = 5 Or _
   DialogSheets(„WASDARFSSEIN“).[D5].Value = 6 Then
   regel = regel & CStr(DialogSheets(“WASDARFSSEIN”).[Dienst5].Value)
   Else
   regel = regel & “0”
   End If
Else
regel = regel & „000“
End If
Cells(akt_row, 49).Value = regel
End Sub
```

Sobald das Dialogfeld für die Eingabe der Mitarbeiterwünsche geschlossen wird, wird aus den eingegebenen, grafischen dargestellten Wünschen wieder die Kodierung aufgebaut, die in Abschnitt 10.6.4. erläutert wurde.

Mitarbeiterwünsche – IV

```
Sub Schaltfläche38_BeiKlick()
,  Stellt alle Werte im Wunschplaner auf 0
DialogSheets(„WASDARFSSEIN“).[D1].Value = 0
DialogSheets(„WASDARFSSEIN“).[D2].Value = 0
DialogSheets(„WASDARFSSEIN“).[D3].Value = 0
DialogSheets(„WASDARFSSEIN“).[D4].Value = 0
DialogSheets(„WASDARFSSEIN“).[D5].Value = 0
DialogSheets(„WASDARFSSEIN“).[T1].Value = 0
DialogSheets(„WASDARFSSEIN“).[T2].Value = 0
DialogSheets(„WASDARFSSEIN“).[T3].Value = 0
DialogSheets(„WASDARFSSEIN“).[T4].Value = 0
DialogSheets(„WASDARFSSEIN“).[T5].Value = 0
DialogSheets(„WASDARFSSEIN“).[R1].Value = 0
DialogSheets(„WASDARFSSEIN“).[R2].Value = 0
DialogSheets(„WASDARFSSEIN“).[R3].Value = 0
DialogSheets(„WASDARFSSEIN“).[R4].Value = 0
```



```
DialogSheets(„WASDARFSSEIN“).[R5].Value = 0
DialogSheets(„WASDARFSSEIN“).[Dienst1].Value = 0
DialogSheets(„WASDARFSSEIN“).[Dienst2].Value = 0
DialogSheets(„WASDARFSSEIN“).[Dienst3].Value = 0
DialogSheets(„WASDARFSSEIN“).[Dienst4].Value = 0
DialogSheets(„WASDARFSSEIN“).[Dienst5].Value = 0
End Sub
```

Wird die Schaltfläche „Löschen“ im Dialogfeld „Mitarbeiter – Planung“ geklickt, so werden alle Kombinationsfeld und Auswahlboxen auf Null zurückgesetzt. Da die Übergabe der Wünsche an die Excel-Tabelle erst mit einem Klick auf „OK“ ausgelöst wird, ist hier eine Übertragung an die Tabelle selbst nicht notwendig.

Teamsitzung zurück

```
Sub team_zurueck()
Dim antwort As String
antwort = MsgBox(„Achtung, soll der interner Merker für die Teamsitzung zurückgesetzt
„ _
& „werden ? Trifft das Programm erneut auf das Schlüsselwort in der Zeile 8, so wird „
_
& „erneut die Kalkulation einer Teamsitzung angeboten.“, vbYesNo, „Frage!“)
If antwort = „7“ Then Exit Sub
ActiveSheet.Unprotect
Range(„AR4“).Value = 0
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
Range(“D9”).Activate
End Sub
```

Im Konzept von DPE3 wird davon ausgegangen, dass nur eine Teamsitzung im Monat durchgeführt wird. Somit erfolgt die Analyse der Zeile 8 auf das Schlüsselwort nur einmalig, bis in der Zelle AR4 ein Wert ungleich 0 steht. Mit dem Klick auf die Schaltfläche „Teamsitzung zurück“ wird lediglich dieser Marker wieder auf Null gesetzt.

Nachwachenmerker auf 0 setzen

```
Sub wache_auf_null()
Dim antwort As String
antwort = MsgBox(„Achtung, dieser Vorgang löscht den Wachenmerker, sprich Sie wissen „ _
& „künftig nicht mehr, wieviele Nachtwachen der einzelne Mitarbeiter innerhalb des „ _
& „aktuellen Plans und des laufenden Quartals erarbeitet hat. Wollen Sie wirklich „ _
& „diese Werte für alle Mitarbeiter auf ‚0‘ setzen ?“, vbYesNo, „Frage!“)
If antwort = „7“ Then Exit Sub
ActiveSheet.Unprotect
Range(“AX9,AX12,AX15,AX18,AX21,AX24,AX27,AX30,AX33,AX36,AX39,AX42,AX45,AX48,
AX51,AX54,AX57,AX60,AX63,AX66,AX69,AX72”).Select
Selection.Value = 0
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
Range(“D9”).Activate
End Sub
```

Für die quartalsweise Feststellung, ob ein Mitarbeiter die kleine beziehungsweise große Nachtwachenzulage erhält, läuft in DPE3 ein Nachtwachenzähler mit, der auf ein Wechsel des Quartals mit einem Rücksetzen reagiert. Sollte es nun einmal notwendig sein, können alle Zähler mit einem Klick auf die Schaltfläche „Wachenmarker auf 0 setzen“ komplett für alle Mitarbeiter zurückgesetzt werden.

Wünsche entfernen

```
Dim antwort As String
antwort = MsgBox(„Achtung - Sie sind im Begriff alle hinterlegten Mitarbeiterwünsche „ _
& „zu entfernen. Soll dieser Vorgang fortgesetzt werden ?“, vbYesNo, „Frage!“)
If antwort = „7“ Then Exit Sub
ActiveSheet.Unprotect
Range(“AW9,AW12,AW15,AW18,AW21,AW24,AW27,AW30,AW33,AW36,AW39,AW42,AW45
,AW48,AW51,AW54,AW57,AW60,AW63,AW66,AW69,AW72”).Select
Selection.Value = “000,000,000,000,000”
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
Range(“D9”).Activate
```

Das ist die Schaltfläche für den zornigen Stationsleiter: Alle hinterlegten Wünsche für Mitarbeiter werden entfernt. Funktionell entspricht dies einem kompletten Eintrag von 5 x 000 mit Kommata unterteilt.

Parameterübernahme

Die Parameterübernahme ist eine notwendige Funktion, da ja kein Datenbankmodell oder ähnliches für den DPE3 verwendet wird, sondern lediglich eine einzige Datei für Microsoft Excel. Um eine neue Version zu erhalten ist es also notwendig die Informationen der alten Fassung in die neue Datei zu übernehmen. Die Übernahme wird dabei auf die Definitionsparameter und die Namen und Einstellungen der Mitarbeiter beschränkt. Eine Übernahme des Planinhalts wird nicht vorgenommen – da in den meisten Fällen auch nicht sinnvoll. Dieser Programmabschnitt ist das Ergebnis eines Makromitschnitts:

```
Sub para_uebernahme()
Dim antwort As String
Dim dversion As String
antwort = MsgBox(“Damit das hier klappt, muss eine ursprüngliche Planversion auf C:\ mit“ _
& „dem Namen ALT.XLS liegen. Die Übernahme funktioniert nur innerhalb der Version III. Sie „ _
& „müssen aktuell in der Datei NEU sein. Im Zweifelsfall lassen Sie die Finger von dem
Befehl.“, vbYesNo, „WICHTIG!“)
If antwort = „7“ Then Exit Sub
ActiveSheet.Unprotect
  Columns(„AL:IV“).Select
  Selection.EntireColumn.Hidden = False
  ActiveWindow.ScrollColumn = 1
  Workbooks.Open Filename:=“C:\ALT.xls“
  Windows(„ALT.XLS“).Activate
  Worksheets(„DEFINITIONEN“).Activate
  dversion = Left$(Cells(2, 3), 1)
  If Left$(Range(„E18“).Value, 10) = „Version 2.“ Then dversion = “2”
  If dversion <> “3” And dversion <> “2” Then
  MsgBox “Es konnte nicht sichergestellt werden, dass die Version in der Datei ALT.XLS zur
Version “ _
  & “im aktuellen Plan passend ist. Der Vorgang wird aus Sicherheitsgründen abgebrochen“,
vbCritical, „Fehler!“
  Exit Sub
  End If
  Worksheets(„PLAN“).Activate

  Range(„A9:A74“).Select
  Selection.Copy
  Windows(„NEU.XLS“).Activate
  Range(„A9“).Select
  Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks:=False,
Transpose:=False
  If dversion = “2” Then
  Sheets(“PLAN”).Select
  Columns(“AM:IV”).Select
  Selection.EntireColumn.Hidden = True
  ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
  MsgBox “Übernahme der Mitarbeiterstammdaten aus der Version 2 vollständig.”, vbIn-
formation, „OK“
  Exit Sub
  End If
  Windows(„ALT.xls“).Activate
  Application.CutCopyMode = False
  ActiveSheet.Unprotect
  Columns(„AL:IV“).Select
  Selection.EntireColumn.Hidden = False
  Range(„AW8:AX72“).Select
  Selection.Copy
  Windows(„NEU.XLS“).Activate
  Range(„AW8“).Select
  Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks:=False,
Transpose:=False
  Windows(“ALT.xls”).Activate
  Worksheets(“DEFINITIONEN”).Activate
  Range(“B3:D136”).Select
  Application.CutCopyMode = False
  Selection.Copy
  Windows(“NEU.XLS”).Activate
  Worksheets(“DEFINITIONEN”).Activate
  Range(“B3”).Select
  Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks:=False,
Transpose:=False
  Windows(“ALT.xls”).Activate
  Worksheets(“DEFINITIONEN”).Activate
  Range(“A145:C153”).Select
  Application.CutCopyMode = False
  Selection.Copy
  Windows(“NEU.XLS”).Activate
  Worksheets(“DEFINITIONEN”).Activate
  Range(“A145”).Select
  Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks:=False,
Transpose:=False
  Sheets(“PLAN”).Select
  Columns(“AM:IV”).Select
  Selection.EntireColumn.Hidden = True
  Range(“D9”).Select
  Application.CutCopyMode = False
  ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
  MsgBox “Übernahme der Daten aus der Version 3 vollständig. Bitte prüfen Sie den
  Bereich 138 “ _
  & “ bis 144 und 157 im Bereich Defintionen”, vbInformation, „Muss sein!“
End Sub
```

Dieser Programmabschnitt muss zu jedem Versionswechsel angepasst werden. Aktuell ist die Version 3.204 freigegeben worden, eine Fassung in der anstelle 22 bis zu 30 Mitarbeiter verplant werden können. Dies hatte umfangreiche Änderungen im ausgeblendeten Excel-Tabellenteil zur Folge, die sich auch auf die Parameterübernahme auswirken.

13 Bedienungsanleitung – Dienstplan in Excel Version 3

Im folgenden Kapitel findet sich quasi die Bedienungsanleitung für die Version 3 des Dienstplans in Excel, so wie er im Januar 2004 am Bezirkskrankenhaus Kaufbeuren eingeführt wurde. Eingangs findet sich eine Checkliste, mit deren Hilfe das vorhandene Modell schnell parametriert werden kann. Je nach Kenntnisstand in Microsoft Excel sollte der Benutzer die Einrichtung in weniger als einer Stunde abgeschlossen haben.

Im Internet findet sich die jeweils aktuellste Fassung. Solange die Hauptversionsnummer, hier die Drei, unverändert bleibt, so wird an der Funktionalität nichts verändert. Die Zahlenkolonne hinter dem Punkt der Version gibt Auskunft über die Unterversion. Jede Unterversion steht für eine fehlerbereinigte Fassung.



Da die Entwicklung des Dienstplaners stetig voranschreitet, kann es sein, dass neuere Versionen bereits Funktionen beinhalten, auf die hier im Detail nicht eingegangen wird. Die aktuellste Fassung unterstützt beispielsweise bereits 30 anstelle 22 Mitarbeiter auf einem Plan, was zu einer Umordnung der aktiven Tabellenteile führt.



13.1. Checkliste für den Ersteinsatz

☐ Prüfung der Voraussetzungen (siehe 13.3)

☐ Installation

☐ einmalige Vorbereitung in Excel (siehe 13.4)

Für alle hier festzulegenden Parameter finden sich detaillierte Informationen im Abschnitt „Parametrierung". Diese Auflistung soll die Einrichtung erleichtern, bereits durchgeführte Schritte sind einfach abzuhaken. Die Einrichtung dauert zirka fünfzehn Minuten. Üblicherweise wird im Folgenden EINSCHALTEN mit 1 und AUSSCHALTEN mit 0 (Null, nicht O) eingetragen.

☐ Stationsnamen eintragen
DEFINITIONEN ➔ Feld C3

☐ Dienstformen hinterlegen
☐ DEFINITIONEN ➔ Bereich C18 – C24
☐ DEFINITIONEN ➔ Bereich C4 – D17 und C17
☐ Ist eine Dienstbezeichnung länger als 1 Zeichen?
➔ DEFINITONEN „Sonderparameter" C91 beachten

☐ Mitarbeiter im Wochenturnus vorhanden ?
➔ DEFINITONEN „Sonderparameter" C94 beachten

☐ Entscheidung : Tagesplanung oder Wochenplanung ?
Wird die Wochenplanung gewählt (Unterscheidung nur MO-FR; SA-SO+FE)
so muss der Parameter C75 in den DEFINITIONEN auf 1 gesetzt werden

☐ Sollbesetzungen definieren
DEFINITIONEN ➔ Bereich C35 – C74 oder Bereich C25 – C34

☐ Welche Mitarbeiter zählen in die SOLL/Ist-Besetzung?
DEFINITIONEN ➔ Bereich C76 – C90

☐ Abweichung von Sollstunden
DEFINITIONEN ➔ Bereich C92 – C93
steht in der Standardauslieferung auf 15% und kann unverändert bleiben

☐ Teamsitzung einrichten
DEFINITIONEN ➔ Bereich C95– C102 und D97 (!)
Standardparameter sind eingerichtet, bitte führendes Hochkomma (über der Taste für das Doppelkreuz (#) bei der Eingabe des Stundenwerts beachten)

☐ Mitarbeiterplan einrichten
DEFINITIONEN ➔ Bereich C103 – C106
Achtung: der Plan wird nur dann gedruckt wenn C104 auf 1 steht!

☐ Sonderdienste einrichten
DEFINITIONEN ➔ Bereich C107 – C117
Beispieleinträge sind vorhanden – Eintrag C117 darf nicht verändert werden

☐ Qualitative Prüfung einrichten
DEFINITIONEN ➔ Bereich C137 – C143
Weiter Informationen finden sich im Abschnitt „Qualitative Planprüfung"

☐ Externe Programme
DEFINITIONEN ➔ Bereich C145 – C146
Für die Anzeige der qualitativen Plananalyse wird der Texteditor verwendet.
Eintrag in C145 lautet üblicherweise für Windows 95,98,ME und XP:
c:\windows\notepad.exe
Für Windows 2000 und Windows NT4.0 lautet der Eintrag üblicherweise:
c:\winnt\notepad.exe

Ist Windows auf Ihrem Rechner nicht auf C: installiert ist der Eintrag entsprechend anzupassen.

Die Datei, die für die Analyse erzeugt wird muss einen Namen und einen Speicherort haben. Üblicherweise wird diese als c:\data.txt benannt. Sollten Sie keine Schreibrechte auf C: haben, oder auf einem Terminalserver arbeiten, so empfiehlt sich die Angabe eines individuellen Homo-Verzeichnisses. Kontaktieren Sie gegebenenfalls Ihre Systemadministration. Der Pfad, auf den Ihre „Eigenen Dateien" zeigen, kann in jedem Fall verwendet werden!

☐ Unterschriftenfelder benennen
DEFINITIONEN ➔ Bereich C152 – C153

☐ Jahresdaten hinterlegen
JAHRESDATEN ➔ Bereich B2 bis D18
Beachten Sie hierzu den Abschnitt „Jahresdaten"

☐ Feiertage hinterlegen
JAHRESDATEN ➔ Bereich B20 bis C49
DPE3 ist für das Bundesland Bayern für die Jahre 2004/2005 eingerichtet. Bei der Einrichtung ist sicherzustellen, dass die Liste im kompletten Bereich mit Einträgen aufgefüllt ist. Verwenden Sie einfach Feiertage für weitere Jahre, falls Sie in einem Bundesland mit weniger Feiertage pro Jahr leben sollten.

Die Werte im Bereich C118 bis C136 müssen üblicherweise nicht verändert werden. Die Einrichtung der Stammparameter ist nun abgeschlossen. Diese Werte werden von einem zum nächsten Plan weiterverwendet. Der Plan sollte nun unter einem neuen Namen gespeichert werden.

- Mitarbeiter einrichten
  Beachten Sie dazu den Abschnitt „Definition von Mitarbeitern“ und „Wünsche“

Auch diese Werte werden für künftige Pläne übernommen. Die Dienstplanung kann nun begonnen werden.

13.2 Grundlegende Gedanken

Die Erstellung eines Dienstplans ist ein komplexer Vorgang bei dem neben den Bedürfnissen der Einrichtung (komplette Besetzung, keine unnötige Schaffung von Mehrarbeit oder Überstunden etc.) auch die Bedürfnisse des Personals (Wünsche, gleiche Verteilung der Sollarbeitszeit auf alle Mitarbeiter) berücksichtigt werden müssen.

Persönliche Informationen wie beispielsweise „Gerhard und Thomas können zusammen mit einem Auto in den Nachtdienst fahren“ – oder „Gaby und Thomas zusammen arbeiten zu lassen ist für alle Beteiligten ein negatives Erlebnis“ werden von einem Dienstplanmodell im Computer nicht berücksichtigt. Nicht weil diese Informationen nicht innerhalb eines Datenstroms bearbeitet werden könnten – sie sind nur sehr schwer zu erfassen, kaum in Worte zu fassen und oft von nur sehr kurzer Dauer. Beim vorliegenden Planmodell wird im Grundsatz davon ausgegangen: „Der Dienstplaner / die Dienstplanerin ist schlau und weiß alle diese Gegebenheiten“.
Auch die recht individuelle Auslegung von Berechnungen im BAT – z.B. „Ist doch klar – da zählen nur die ersten drei Stunden als Feiertagsvergütung“ werden derzeit dem menschlichen Dienstplaner aufgebrummt. Der lästige Teil – Ausrechnen der Stunden, Prüfung der Besetzung und eine möglichst gerechte Verteilung zwischen den Kollegen wird im aktuellen Modell berücksichtigt.

Der Mensch hat sich im Laufe seiner Entwicklung an das Phänomen Uhrzeit gewöhnt – im Gegensatz zu den meisten Anzeigen und Werten in unserem Leben, die auf der Zehnerpotenz basieren, ist der Grundwert der Uhrzeit 24 Stunden und 60 Minuten doch eher exotischer. Würde die Uhr in den heutigen Tagen von Informatikern neu erfunden werden so würde es wahrscheinlich eine Uhr mit nur 16 Stunden geben. Im Dienstplaner mit Excel werden Zeitwerte ausschließlich im Format „Stunden:Minuten“ eingegeben. Ein Rückschritt auf das Dezimalsystem wird dem Benutzer erspart.

13.3 Voraussetzungen

Für den korrekten Betrieb des Dienstplan in Excel III (von nun an als DPE3 bezeichnet) ist, wie man dem Namen schon entnehmen kann, Microsoft Excel erforderlich. Es gibt Microsoft Excel in verschiedenen Versionen. Für den Betrieb von DPE3 wird Microsoft Excel in einer Version nach 2000 benötigt. Aktuell entspricht das den Fassungen 2000, XP und 2003 für Microsoft Windows. Der Betrieb unter Microsoft Office für Machos X wurde nicht getestet. Die Leistungsanforderungen an die Hardware werden durch das Betriebssystem Microsoft Windows bereits festgelegt. Für jeden Monat, der in DPE3 geplant wird, ist zirka ein halbes MByte an Speicherplatz zu veranschlagen. Die Monitorauflösung sollte im besten Fall 1024x768 Punkte betragen, die Bildwiederholungsfrequenz sollte in einem Bereich über 85 Hz liegen, um ein ergonomisches, längeres Arbeiten vor dem Computer zu ermöglichen. Der Ausdruck des Mitarbeiterdienstplans geschieht automatisch auf den in Windows definierten Standarddrucker.

Installation von einem Datenträger

Öffnen Sie den Microsoft Explorer über das Startmenü oder über die Tastenkombination „Fenstertaste + E“. Wählen Sie den gewünschten Quelldatenträger, beispielsweise Laufwerk E für das CD-ROM Laufwerk. Ziehen Sie im Explorer die DPE3.XLS Datei in den Ordner eigene Dateien. Es wird empfohlen einen Unterordner anzulegen, der die Dienstpläne aufnimmt – im günstigsten Fall für jedes Jahr ein eigener Ordner. Bewahren Sie den Quelldatenträger an einem sicheren Ort.

Installation über das Internet bzw. Intranet

Laden Sie die aktuellste DPE3 Datei aus dem Internet, einen entsprechenden Download-link finden Sie

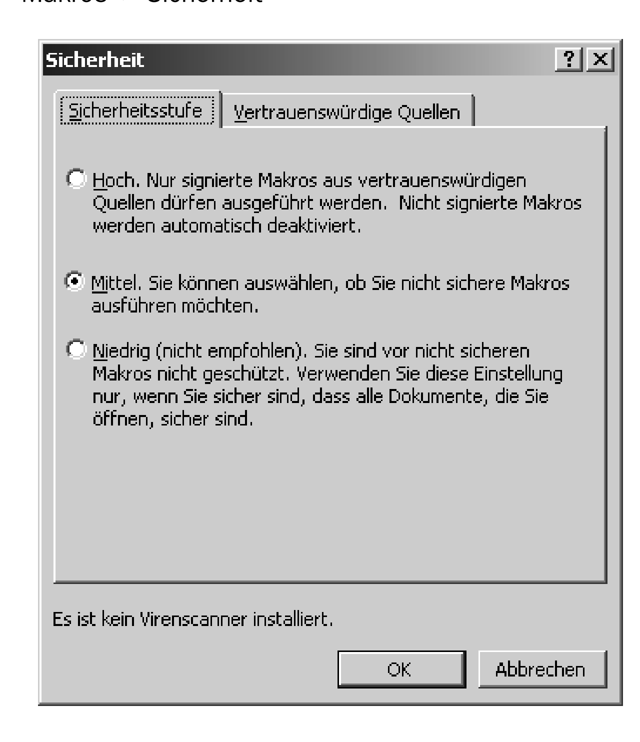
auf der Homepage des Pflegeinstituts (www.pflegeinstitut.de). Die Datei wird in zwei Formen zum Download angeboten. Die eine Fassung als ausführbare Datei im EXE-Format, die andere als ZIP-Datei. Beide Versionen sind mit dem ZIP-Algorithmus komprimiert und müssen vor dem Betrieb entpackt werden. Während die ZIP-Version den Besitz eines eigenen „Entzippers“ (beispielsweise im Microsoft Windows XP-Betriebssystem integriert) voraussetzt, ist in der EXE-Datei der Dekompressionscode mit integriert. Sollte Ihre Firewall den Download von ausführbaren EXE-Dateien verhindern, sollten Sie den Download der ZIP-Version vorziehen. Zum Umgang mit ZIP-Archiven finden Sie auf der Homepage des Pflegeinstituts eine entsprechende Anleitung.

Kopieren Sie die Datei in den Ordner „Eigene Dateien“ und dekomprimieren Sie die Datei mit einem Doppelklick. Es wird empfohlen einen Unterordner anzulegen, der die Dienstpläne aufnimmt – im günstigsten Fall für jedes Jahr ein eigener Ordner. Bewahren Sie den Quelldatenträger an einem sicheren Ort.

13.4 Vorbereitungen in Excel

Starten Sie zunächst einmal Microsoft Excel ohne die soeben vorbereitete XLS-Datei zu öffnen. In der Regel wird Microsoft Excel über das Startmenü gestartet.

Klicken Sie auf Extras -> Makros -> Sicherheit



Setzen Sie den Wert auf „Mittlere Sicherheit“. In den hohen Sicherheitseinstellungen werden Makros und Visual Basic for Applications Programme – beides wird später noch im Detail betrachtet – deaktiviert, sodass weiterführende Techniken nicht zur Verfügung stehen. Sollten Sie nach der Festlegung der Einstellung auf „Mittel“ ein Service Pack, Hotfix oder Patch für Microsoft Office beziehungsweise Excel auf einem Rechner Einspielen, so ist der Wert automatisch wieder auf „Hoch“ eingestellt. Hintergrund dieses Verhaltens ist der Versuch vom Hersteller Microsoft, die Applikationssicherheit gegenüber Trojanern und Programm-Würmern aus dem Internet zu erhöhen.

Um Eingabefehler zu verhindern empfiehlt es sich, die „AutoEingabe“ von Microsoft Excel abzuschalten. Diese Funktion ermöglicht die automatische Vervollständigung von begonnenen Worten in einer angrenzenden Zelle innerhalb einer Spalte oder Reihe. Für die Dienstplanung könnte dies zur Folge haben, dass unterhalb eines eingetragenen Bereitschaftsdienst „BD“ ein einfacher „B“-Dienst kaum einzutragen ist.

Klicken Sie auf Extras -> Optionen -> Bearbeiten

Entfernen Sie das Häkchen von „AutoEingabe für Zellwerte aktivieren“.
Anschließend können Sie Microsoft Excel schließen, beide Änderungen bleiben erhalten (Einschränkungen bezüglich der Makro-Sicherheit siehe oben).

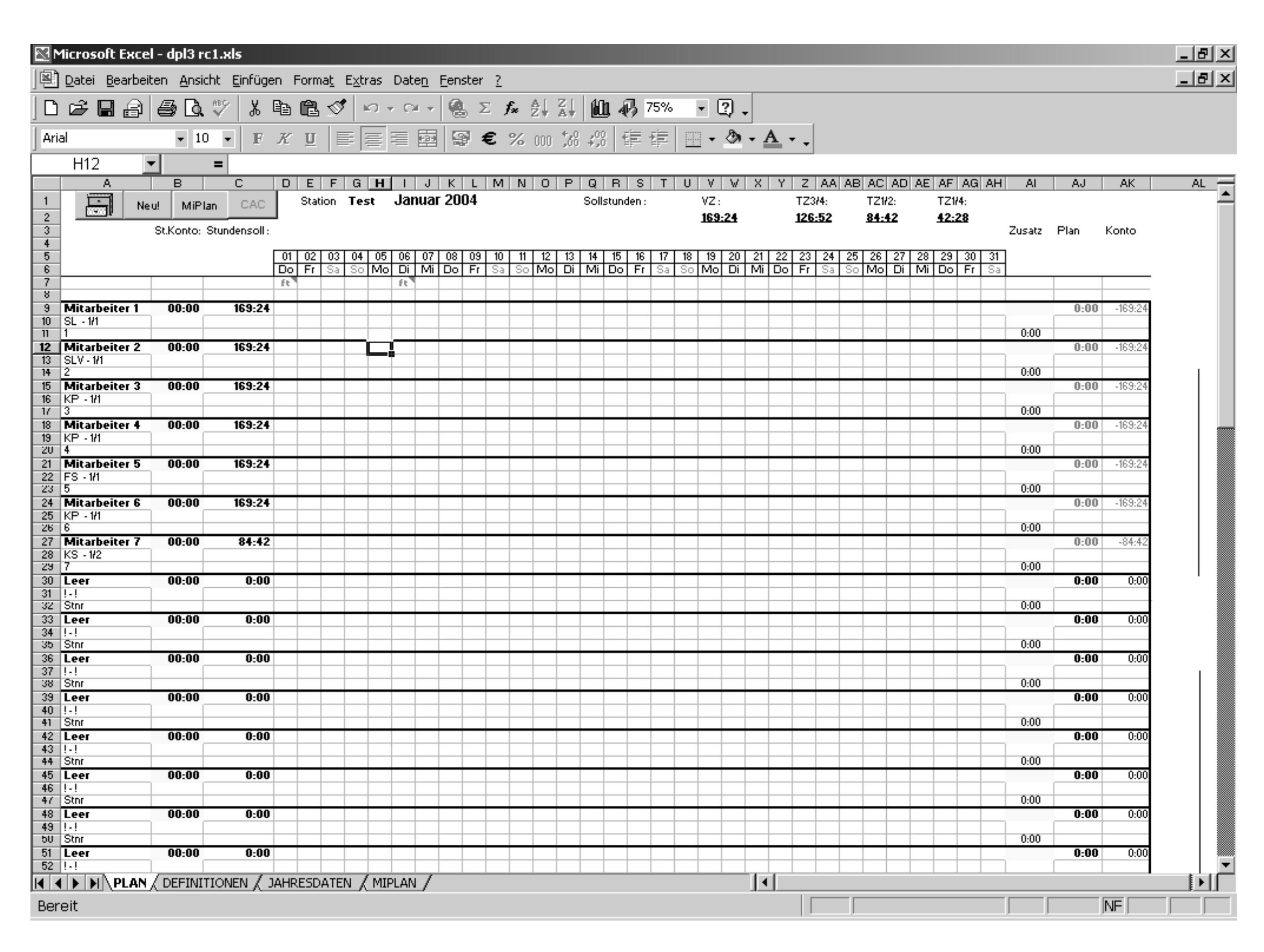
13.5 Starten von DPE3

Starten Sie DPE3 mit einem Doppelklick auf die XLS-Datei. Sollten Sie einen neuen Planungsmonat einrichten wollen, so empfiehlt es sich relativ zeitnah mit dem Befehl Datei -> Speichern unter... die Datei mit einem neuen Dateinamen abzulegen. Sprechende Dateinamen wie „D2_11_04.XLS" machen das Widerauffinden eines älteren Planes einfacher. Bevor Sie mit der Planung beginnen können, werden Sie stets von Microsoft Excel über das Vorhandensein von Makros informiert:



Solange die Makrosicherheit auf „MITTEL" steht, erscheint dieser Dialog. Klicken Sie auf „Makros aktivieren" um mit dem Programm arbeiten zu können. Werden die Makros deaktiviert, so stehen die meisten Funktionen nicht mehr zur Verfügung und der Dienstplan ist unbrauchbar. Das Herabsetzen der Makro-Sicherheit auf „NIEDRIG" hätte zwar zur Folge, dass hier nicht immer die Benutzung von Markos bestätigt werden muss, ein Sicherheitsrisiko für andere Office-Anwendungen entsteht aber gleichzeitig. Empfehlung: Belassen Sie den Wert auf „MITTEL" und bestätigen Sie bei jedem Start mit „Makros aktivieren".

Anschließend erscheint die gewählte Datei. Sollten Sie die ursprüngliche DPE3.XLS Datei ausgewählt haben, so sehen Sie vor sich einen leeren Dienstplan, ohne Parametrierung.



13.6 Funktionelle Bereiche

Die Dienstplanung erleidet, unabhängig vom gewählten Modell, immer die gleiche Krankheit: Platzmangel. Um möglichst viele Funktionen auf geringem Raum unterzubringen wurden die Zellen in verschiedene funktionelle Bereiche unterteilt.

ERSTE ZEILE

Die erste Zeile eines jeden Mitarbeiters (sprich 9,12,15 u.s.w.) trägt das geplante Kürzel für den Dienst, beispielsweise U für Urlaub oder W für Wache. Jede Einrichtung pflegt unterschiedliche Kürzel, diese lassen sich im Fenster DEFINITIONEN festlegen – doch dazu mehr im Abschnitt „Parametrierung". Ob die Kürzel in großen oder kleinen Buchstaben hinterlegt und geschrieben werden ist dem Benutzer überlassen, es sollte jedoch konsequent die gleiche Methode verwendet werden.

Eingaben von berechnungsrelevanten Dienstkürzeln führen automatisch zur Aufaddierung in den Spalten AJ und AK.

Mit einem Doppelklick auf einen zu planenden Tag wird der Planungshelfer für den jeweiligen Mitarbeiter geöffnet. Zum einen sehen Sie hier hinterlegte Wünsche des Mitarbeiters, zum anderen können Sie Sonderdienstformen eingeben. Weitere Informationen finden im Abschnitt „Planungshelfer".

ZWEITE ZEILE

Die zweite Zeile innerhalb des Planungsbereichs dient für freie Einträge, die zu keiner Berechnung führen. Eine Ausnahme stellt die Buchstabenfolge „BD" dar, die vom Programm als Bereitschaftsdienst interpretiert werden und ohne Berechnung auf den Mitarbeiterplan übernommen werden.

DRITTE ZEILE

Die dritte Zeile dient in erster Linie zur Eingabe von Zusatzzeiten im Format HH:MM. Soll beispielsweise zu einem Dienst, der an einem Feiertag stattfindet die Verdoppelung der Stunden berechnet werden, so lassen sich hier 07:42 oder 08:00 als Stundenwert in der dritten Zeile eintragen. Alle Zusatzwerte ebenfalls direkt in den Spalten AJ und AK automatisch aufaddiert. Die Eingabe von negativen Werten ist in der dritten Zeile nicht möglich und führt zu einer fehlerhaften Anzeige von Doppelkreuzen. Das Zwischenergebnis aller zusätzlichen Stundenwerte wird in der Spalte AI in der dritten Zeile angezeigt.

STUNDENÜBERTRAG

Der Stundenübertrag (positiv oder negativ) aus dem Vormonat wird in der Spalte B im gelb eingefärbten Feld eingetragen. Auch hier gilt das Format HH:MM. Sollen beispielsweise zehn Stunden und fünfzehn Minuten übergeben werden, so werden diese ohne führendes Vorzeichen in dieses Feld eingetragen. Negative Überträge werden mit einem führenden Minuszeichen eingegeben.

Ein Stundenübertrag wirkt sich auf den weiteren Kontenverlauf (Spalte AK) aus, nicht aber auf die Stundenanzeige für diesen Monat. Somit soll dem Planer die Erfüllung der Sollstunden des aktuellen Monats vordergründig dargestellt werden.

STUNDENSOLL

Das Stundensoll in Spalte C ergibt sich aufgrund der Definition des Anstellungsverhältnisses (1/1 = Vollzeitkraft, 3/4 = Teilzeitkraft mit 75%, 1/2 = Teilzeitkraft mit 50% und 1/4 = Teilzeitkraft 25%) und des laufenden Monats, der im Fenster „JAHRESDATEN" definiert ist. Ein weiterer Wert im Anstellungsverhältnis ist „WT" (Wochenturnus) – wie sich das Verhältnis des Wochenturnus gegenüber den anderen Anstellungen darstellt, wird über das Fenster „Definitionen" über einen eigenen Parameter gesteuert. Die automatisierte Berechnung des Wochenturnus ist ein Feature für eine folgende Version.

Nun kann es natürlich sein, dass eine Vollzeitkraft nicht das volle Plansoll erfüllen muss, beispielsweise wenn das Dienstverhältnis erst in der Mitte des Monats beginnt. Um den Standardwert der Vorlage zu übergehen kann in der Zelle unterhalb des STUNDENSOLLS im Format HHH:MM ein frei definierbarer Wert eingetragen werden.

ZUSATZ

In der Spalte AI finden sich für jeden Mitarbeiter gelbe Felder. In diese Fenster können positive Stundenwerte im Format HHH:MM eingegeben werden, ohne das diese Zeitwerte einem Dienst zugeordnet werden müssen. Dies ist das einzige Feld in dem einfach Werte eingegeben werden können, um auf die Ergebnisse in den Spalten AJ und AK Einfluss zu nehmen.

ZEILE 8

Diese unterhalb der automatisierten Anzeige der Feiertage freie Zeile dient für die Aufnahme von Zusatzinformationen, wie beispielsweise ein an diesem Tag von der Station zu besetzender Bereitschaftsdienst. In dieser Zeile ist der automatisierte Aufruf der Teamsitzungsberechnung hinterlegt, weitere Informationen finden sich im Abschnitt „Teamsitzung“.

Definition von Mitarbeitern

Mit einem Doppelklick auf das Bezeichnerfeld für den Mitarbeiter wird das Dialogfeld zur Definition von Mitarbeiterstammdaten angezeigt.



Der Name des Mitarbeiter sollte im Format Nachname, Freizeichen und erster Buchstabe des Vornamens gefolgt von einem Punkt eingegeben werden (beispielsweise Müller K.). Die Dienstform ist auszuwählen und hat Einfluss auf die SOLL/Ist-Stellung im Plan. Die Dienstformen sind im Programm fest verankert und lassen sich nicht verändern.

AP	Altenpfleger
AS	Altenschwester (Altenpflegerin)
FP	Fachpfleger
FPK	Fachpflege Praktikant (Hospizeinsatz)
FS	Fachschwester
HEP	Heilerziehungspfleger / Heilerziehungspflegerin
KP	Krankenpfleger
KPH	Krankenpflegehelfer / Krankenpflegehelferin
KPS	Krankenpflegeschüler / Krankenpflegeschülerin
KS	Krankenschwester
SD	Sicherheitsdienst (Forensik) / Sozialdienst
SL	Stationsleiter / Stationsleiterin
SLV	Stationsleiter / Stationsleiterin Verterter/In
SON	Sonstiges Anstellungsverhältnis (z.B. Schüler, FSJ)
ZDL	Zivildienstleistender

Die Kartennummer kann als Stempelkartennummer oder als sonstige eindeutige Nummer (beispielsweise Personalnummer) interpretiert werden. Funktionell hängt aktuell an dieser Nummer keine Funktion. In einer künftigen Version wird über diese Nummer die Wiedererkennung über mehrere Plandateien zu

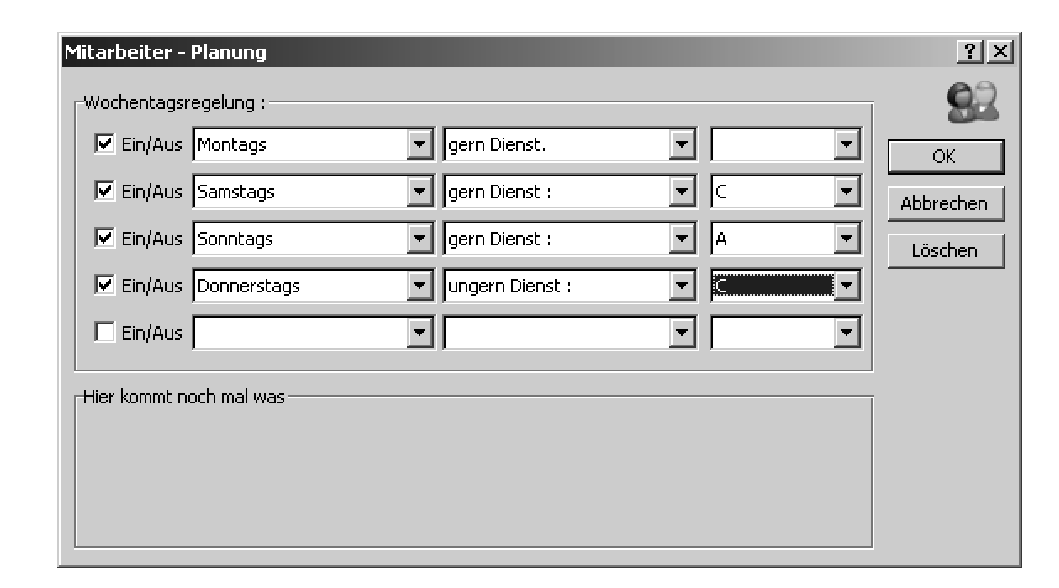
Analysezwecken benötigt (falls beispielsweise die Anordnung der Mitarbeiter auf dem Dienstplan verändert wurde).

Unter Anstellung wird die Stundensollstellung im Anstellungsverhältnis festgelegt. Eine frei definierbare Sollstellung ist aktuell nur über das Feld Stundensoll möglich. Weitere Informationen dazu finden sich im Abschnitt „Funktionelle Bereiche – Stundensoll“.

Wird ein Mitarbeiter in der Planung nicht mehr benötigt kann der Eintrag mit der Schaltfläche „Löschen“ entfernt werden. DPE3 ignoriert alle Mitarbeiter die entweder „Leer“ heißen, oder gar keinen Namen besitzen.

13.7 Wünsche

Mit der Schaltfläche „Wünsche“ wird das Dialogfeld zur Abbildung von Dienstvorlieben der Mitarbeiter angezeigt.



Insgesamt lassen sich bis zu fünf Vorlieben definieren. Auf der rechten Seite wird die Regel aktiviert oder deaktiviert (beim nächsten Aufruf des Fensters ist der Eintrag leer). Der Wochentag zuzüglich des Feiertags wird im nächsten Feld ausgewählt. Die nächste Auswahl betrifft in sechs Abstufungen verschiedene Vorlieben:

... gar keinen Dienst.
... ungern Dienst.
... gern Dienst.
... am liebsten Dienst.
... gern Dienst <DIENSTFORM>
... ungern Dienst <DIENSTFORM>

Die letztgenannten Vorlieben erfordern die Eingabe einer Dienstform, die automatisch aus den Definitionen übernommen wird. Sollten widersprüchliche Regeln aufgebaut werden, so gilt die zuletzt genannte Regel als die dominierende für das weitere Verfahren. Die gewählten Vorlieben werden auch auf einen künftigen Planungsmonat übernommen. Auch die Umstellung der Mitarbeiter in der Reihenfolge auf dem Dienstplan wirkt sich negativ auf diese Werte aus – sie werden nicht automatisch dem neuen Mitarbeiter zugeordnet (das nennt sich Verbesserungspotential, wie man dem Dialogfeld insgesamt ansehen kann).

Die Vorlieben des Mitarbeiters werden in der qualitativen Planprüfung (CAC) und im Planungshelfer berücksichtigt.

13.8 Planungshelfer

Dienste können durch Eingabe in die erste Zeile für den Mitarbeiter einfach per Tastatur eingetragen werden. Die Berechnung der hinterlegten Zeitwerte geschieht automatisch. Methoden, die Microsoft Excel bietet, beispielsweise das Auffüllen mit dem Auffüllkästchen stehen auch im DPE3 zur Verfügung.

Eine zweite Möglichkeit Dienste für den Mitarbeiter einzutragen ist das Dialogfeld des „Planungshelfer“, welches durch einen Doppelklick auf den entsprechenden Tag im Plan geöffnet wird.



Im oberen Fensterbereich können Standarddienste wie in der Definition festgelegt eingetragen werden – der Multiplikator sorgt für eine entsprechende Wiederholung des Dienstes (die Werte des Multiplikators werden ebenfalls über die Definition gesteuert).

Die Schaltfläche „Keinen Dienst“ entfernt alle Diensteinträge für diesen Tag. Dies ist besonders dann wichtig, sobald Sonderdienste eingetragen wurden. Nur mit Hilfe dieser Schaltfläche lassen sich alle Einträge wieder entfernen.

Sonderdienstformen, beispielsweise Gruppen, werden gesondert im Plan dargestellt. In der Standardauslieferung wird die Planung eines Sonderdienstes mit drei Ausrufezeichen symbolisiert. Verweilt der Mauszeiger auf dem !!! im Plan so erscheint eine kleine Sprechblase, die den geplanten Sonderdienst anzeigt. Die Einträge des Sonderdienstes werden im Mitarbeiterplan ebenfalls angezeigt. Sonderdienstformen werden über die Definitionen gesteuert und besitzen was die Dauer, Beginn und Ende angeht feste Werte (beispielsweise eine Gesprächsgruppe von 17:30 – 19:00 Uhr mit 1 ½ Anrechnung).



Neben den festen Sonderdienstformen lassen sich auch individuell bei Bedarf komplett eigene Sonderdienste abbilden. Dazu wird der dritte Abschnitt des Fensters verwendet:

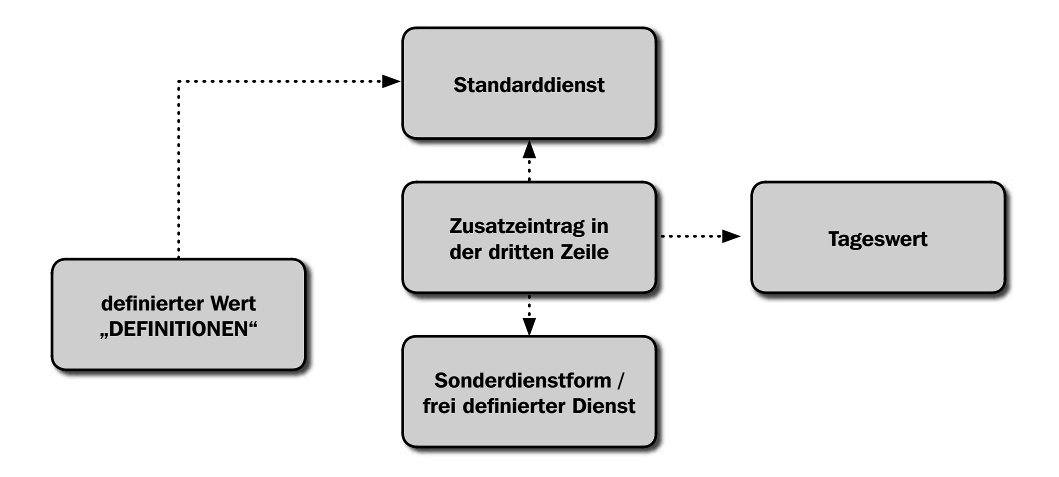


Bei der Verwendung dieser Funktion ist folgendes zu beachten: Die Berechnung des Zeitwerts im Format HH:MM muss manuell durch den Benutzer geschehen. Die Zeitangabe hinter dem Titel der freien Definition muss wie auf der Abbildung durchgeführt werden. Wird von der Konvention abgewichen ist das Programm nicht in der Lage den frei definierten Dienst korrekt in den Mitarbeiterplan einzutragen.

Symbolisiert wird der frei definierte Dienst wie eine Sonderdienstform.

Wird ein Sonderdienst bzw. ein frei definierter Dienst eingetragen und anschließend ein Standarddienst eingetragen, so bleiben die Einträge für den Sonderdienst dennoch erhalten. Somit lassen sich ein Gruppentermin und ein Standarddienst gleichzeitig in den Mitarbeiterplan eintragen. Eine Beschränkung liegt aktuell darauf, dass nur ein einziger Sonderdienst, frei definierter Dienst und ein Zusatz wie Teamsitzung möglich ist.

Zur Berechnung des Tageswerts geht DPE3 wie folgt vor:



Im unteren Fensterbereich findet sich die Rubrik „Aus den Wünschen“. Entsprechend der Vorlieben des Mitarbeiters werden hier Wunschvorstellungen dargestellt. Wie in der Abbildung gezeigt symbolisiert ein „gelber Smiley“ mit dem Text „Gern Dienst“, das der Mitarbeiter an diesem Tag ganz gern Dienst machen würde.

Folgende Tabelle zeigt die möglichen Vorlieben symbolisch:

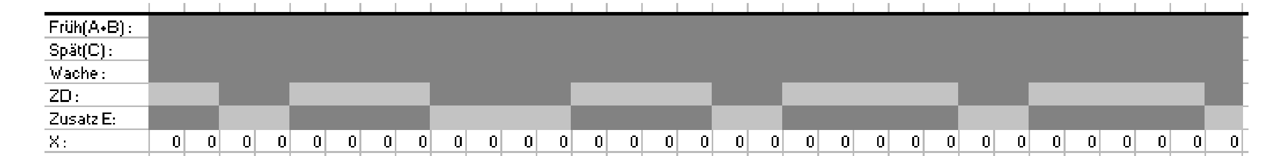
	Grüner Smiley	Generell am liebsten Dienst an diesem Tag.
	Gelber Smiley	Generell gern Dienst an diesem Tag.
	Grüner „Sad“ Smiley	Generell ungern Dienst an diesem Tag.
	Roter „Sad“ Smiley	Generell kein Dienst an diesem Tag.
	Herzsymbol	Sehr gerne die angezeigte Dienstform an diesem Tag.
	Handsymbol	Nicht die angezeigte Dienstform an diesem Tag.

Die Vorlieben des jeweiligen Mitarbeiters werden im Planungsbereich seit Version 3.20 in der Statusleiste direkt angezeigt:

Dienst++	der gewählte Mitarbeiter wünscht sich an diesem Tag einen Dienst
Dienst+	der gewählte Mitarbeiter macht an diesem Tag gern Dienst
Dienst-	der gewählte Mitarbeiter möchte ungern Dienst haben
Dienst--	der gewählte Mitarbeiter möchte an diesem Tag keinen Dienst haben
Dienst : A	der gewählte Mitarbeiter möchte gern an diesem Tag einen A-Dienst
ungern Dienst : W	der gewählte Mitarbeiter möchte

13.9 Soll-Ist-Besetzung

Am unteren Rand der Tabelle findet sich eine farbliche Darstellung der Soll/Ist-Besetzung.



Rot steht für nicht komplett besetzte Schicht.
Grün steht für ausreichend besetzte Schicht.
Orange steht für mehr als ausreichend besetzte Schicht.

„x“-Reihe zählt die im Plan geschriebenen „X“ für die Vorplanung, oder für sonstige Zwecke. Der „X“-Dienst wird nicht gezählt oder berechnet. Es ist eine einfache Unterstützung.

Ziel der Planung sollte sein, dass nach der Planung keine rote Fläche mehr sichtbar ist - - sprich alle Dienste sind besetzt. Haben Sie beispielsweise den „Zusatz E“ Dienst auf Null in den Definitionen gesetzt, so erscheint die Reihe komplett in Grün.

Ob ein Mitarbeiter mit in die Sollbesetzung zählt ist abhängig von der Definition.

Mit der Version 3.20 wurde die Soll-Ist Besetzung direkt in die Statusleiste von Microsoft Excel integriert. So hat der Dienstplaner stets den Bedarf von Schichten im Blick.



Offene Dienste werden hier als positiver Wert angezeigt – im obigen Beispiel werden somit 2x A, C und W und 1 B-Dienst benötigt. Erscheint die Anzeige mit „OK!“ so sind alle Sollstellungen für den Tag erfüllt.

13.10 Neuer Monat

Um einen neuen Planungsmonat einzurichten muss lediglich die Schaltfläche „Neu!“ angeklickt werden. Welcher Monat erzeugt werden soll wird über ein Dialogfeld festgelegt. Die Stammdaten des Planungsmonats, die Sollstunden und die Feiertage des Monats werden aus den Definitionen entnommen.

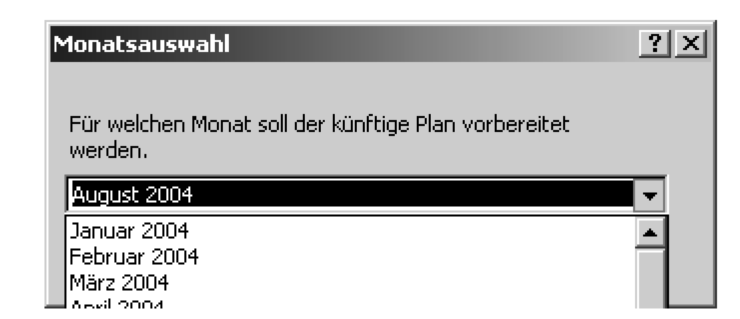
Da der aktuelle Plan durch die Vorbereitung für den neuen Monat gelöscht wird, sollten Sie vor diesem Schritt auf jeden Fall den Plan unter einem neuen Namen abspeichern. Um ein versehentliches Löschen des Plans zu verhindern erscheint zunächst ein Dialogfenster in dem der neue Plan bestätigt werden muss.



Nach der Bestätigung für den neuen Monat erscheint ein weiteres Dialogfenster in dem die Übernahme des aktuellen Stundenkontos in den neuen Plan veranlasst werden kann. Dies bringt nur dann Sinn, wenn die Stundenkonten, die im Altmonat vorhanden sind auch zu 100% mit der Realität übereinstimmen. Verwendet die Einrichtung eine Zeiterfassung so sind diese Werte gegebenenfalls händisch einzutragen.

Wird dieses Dialogfeld mit JA beantwortet, so werden die Stundenkonten in den neuen Plan übertragen. Wird dieses Feld mit NEIN beantwortet, so werden die Stundenkonten auf 00:00 gesetzt. Mit Abbrechen wird die Neuerstellung des Plans generell abgebrochen.

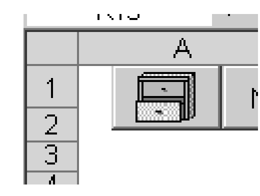
Anschließend wird der Plan bereinigt – dieser Vorgang kann je nach Rechenleistung der Maschine bis zu 30 Sekunden dauern.



Der zu erstellende Monat muss nur noch ausgewählt werden, die Sollstundenwerte und Feiertage werden eingerichtet werden. Dieser Vorgang sollte nur wenige Sekunden dauern.

13.11 Ein- und Ausblendung

Je nach Monitorgröße kann es zur Planung nützlich sein die Zusatzzeile für die Stunden (dritte Zeile) nicht angezeigt zu bekommen. Mit der Schaltfläche „Ein- und Ausblendung“ wird dies vorgenommen.



13.12 Teamsitzung

Die Teamsitzung wird im DPE3 besonders behandelt. Zum einen finden sich einige Parameter in den Definitionen, mit denen der Umfang der Teamsitzung festgelegt wird zum anderen wird die Erstellung vom Programm automatisiert. Empfehlenswerterweise wird die Teamsitzung als eine der letzten Schritte bei der Planung eingetragen, da nur so gewährleistet werden kann, das DPE3 möglichst viel Schreibarbeit einspart.

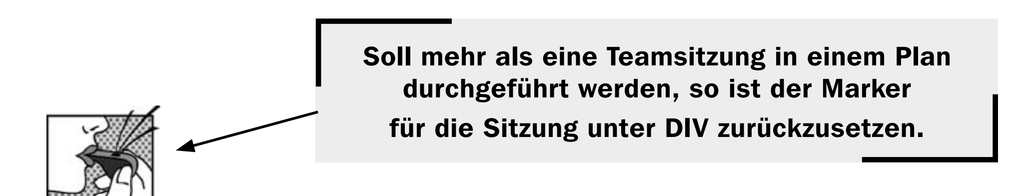
Die Erstellung eines automatischen Eintrags für die Teamsitzung wird über die Zeile 8 mit einem Schlüsselwort begonnen, welches in der Standardparametrierung „Team“ lautet. Wird an einem Tag in der Zeile 8 „Team“ eingegeben, so erscheint ein Dialogfenster. Da DPE3 davon ausgeht, dass nur eine Teamsitzung pro Monat durchgeführt wird, erscheint dieses Dialogfenster entsprechend nur einmalig. Sollte hier für Ihre Einrichtung eine Anpassung erforderlich sein, finden sich die notwendigen Informationen im technischen Handbuch.

Im Folgenden sollen die Parameter der DEFINITIONS – Tabelle zum Thema „Teamsitzung“ näher betrachtet werden:

Teamsitzung :	Schlüsselwort in Zeile 7	Team	Standard : Team
	Diensteintrag für Mitarbeiter	TM	Standard : TM
	Teamstundenwert (+ führendes Hochkomma !)	00:45	19:00 - 19:45
	Dienst mit Muster hinterlegen	1	Standard : 1
	Teamsitzung in der Spätdienstzeit ?	1	Standard : 1
	Nachtschichtler kommt zur Teamsitzung ?	1	Standard : 1
	KPS kommt zur Teamsitzung ?	0	Standard : 0
	ZDL kommt zur Teamsitzung ?	1	Standard : 0

Das Schlüsselwort in Zeile 8 gibt an mit welchem Wort der Automatismus der Teamsitzungsberechnung eingeleitet werden soll. Die Standardeinträge finden sich im roten Bereich auf der rechten Seite, lediglich die Zelle mit dem Uhrzeitenwert findet sich in Spalte D und ist ein Produktivwert.

Der Diensteintrag für Mitarbeiter wird sowohl auf dem Dienstplan als auch auf dem Mitarbeiter-Plan so angedruckt, wie hier eingestellt. Empfehlenswerterweise sollte die Buchstabenkombination nicht länger als zwei Zeichen sein, da der Platz auf dem Dienstplan sehr begrenzt ist.



Der anzurechnende Stundwert im Format HH:MM ist mit einem führenden Hochkomma (Großschreibtaste + #) einzugeben. So wird aus einer halben Stunde Teamsitzung der Eintrag 00:30. In der Spalte D wird der Zeitraum im Format HH:MM LEERZEICHEN DOPPELPUNKT LEERZEICHEN HH:MM eingegeben. Nur wenn das Format korrekt eingehalten wird, ist die korrekte Übernahme in den Mitarbeiterplan gewährleistet.

Im Dienstplan lässt sich der Tageseintrag mit einem leichten Punktmuster andrucken, sofern der Parameter auf 1 gestellt wird.

Findet die Teamsitzung im Zeitraum des Spätdienstes statt, so ist dies hier mit einer 1 einzutragen. Das hat zur Folge, dass dem Spätdienst der festgelegte Stundenwert für die Teamsitzung nicht aufaddiert wird. Erscheint der Nachtdienst ebenfalls zur Teamsitzung, so ist dies mit dem Wert 1 einzutragen, der Stundwert für die Teamsitzung wird aufaddiert. Ob Zivildienstleistende und Krankenpflegeschülerinnen bzw. Krankenpflegeschüler zur Teamsitzung kommen, wird ebenfalls an dieser Stelle gesteuert.

13.13 Jahresdaten

In den Jahresdaten (eigenes Register für eine Tabelle unten rechts) werden neben den Stundensollwerten für die Monate auch die Feiertage hinterlegt. Da Feiertage in der Bundesrepublik in jedem Bundesland anders geartet sind, ist hier für den Einsatz außerhalb von Bayern auf jeden Fall ein Parametrierungsbedarf. Die Übersicht von Feiertagen finden sich im Internet (einfach mal „Feiertage“ googlen). Auch eigene, einrichtungsspezifische Feiertage lassen sich hinterlegen.

Die Planmonate werden im Bereich B2-D18 hinterlegt.

Januar 2004	169:24	1/1/2004

In der Spalte B wird der Name des Monats ausgeschrieben als Text eingetragen, die Sollstunden für Vollzeitkräfte für diesen Monat werden im Format HHH:MM eingegeben – die Berechnung der Stundenwerte für Teilzeitkräfte wird vom Programm automatisch vorgenommen. Der Eintrag in der Spalte D ist der erste Tag des Monats im amerikanischen Format MM/TT/JJJJ. Dieser Wert ist für den Aufbau eines neuen Plans wichtig.

Im Bereich B20-C49 sind die Feiertage hinterlegt. Der Bereich ist für zwei Jahre im Bundesland Bayern ausgelegt. Sollten in Ihrem Bundesland weniger Feiertage in
zwei Jahren existieren, so ist der Bereich mit den Feiertagen aus dem folgenden Jahr aufzufüllen.

Sowohl bei den hinterlegten Monaten, als auch bei den Feiertagen ist es nicht notwendig mit dem Anfang des Jahres zu beginnen. Der Plan ist für den Zeitraum von siebzehn Monaten ohne Änderungen an den Jahresdaten eingerichtet. Sollte diese Vorbereitung nicht ausreichen, so finden sich im technischen Handbuch Hinweise auf die Programmiertechnik um diesen Wert zu erhöhen. Geplant für eine zukünftige Version ist die Algorithmen der Feiertage und Stundensollwerte zu hinterlegen – problematisch ist dann die Anpassung an einrichtungsspezifischen Regelungen, die einem Feiertag ähnlich sind.

13.14 Mitarbeiterplan

Der Mitarbeiterplan ist eine gesonderte Ausgabe des Dienstplans, in der für jeden Mitarbeiter ein eigener, individueller Plan ausgegeben wird. Neben den Kollegen, mit der sich der jeweilige Mitarbeiter die Schicht teilt, werden Sonderdienste, frei definierte Dienstformen, Teamsitzung und Bereitschaftsdienste angezeigt. Entgegen des üblichen Dienstplans, der horizontal angelegt ist, wird der Mitarbeiterplan vertikal ausgegeben.

Über das Register MIPLAN lassen sich die Optik des Mitarbeiterplans und der Druckbereich frei einstellen. Die jeweiligen Informationen werden zur Laufzeit vom Programm automatisch eingetragen.

Ausgegeben wird der Mitarbeiterplan mit einem Klick auf die Schaltfläche „MiPlan“ im oberen Planbereich. Da der Mitarbeiterplan nach Erstellung automatisch auf den Windows-Standarddrucker ausgegeben wird, ist die Leistung des Druckers für die Ablaufgeschwindigkeit mitentscheidend. Damit der Druck wirklich gestartet wird ist in den DEFINITIONEN der Parameter „Drucker wirklich starten (DEBUG Parameter)“ (Zelle C104) auf 1 zu stellen. Ob Dienstzeiten angedruckt werden und der Bereitschaftsdienst aus der zweiten Zeile des Planungsbereichs in den Mitarbeiterplan übernommen werden soll, ist ebenfalls in diesem Bereich der DEFINITIONEN-Tabelle einzustellen.

13.15 Qualitative Plananalyse – CAC

Der Commissaire des Comptes (zu Deutsch Rechnungsprüfer) ist ein Konzept zur Analyse eines Dienstplans auf die Qualität. Ob dieses Verfahren überhaupt eingesetzt wird und mit welchen Punktwerten, wird über die DEFINITIONEN-Tabelle gesteuert.

Folgende Gedanken stehen hinter dem CAC:

Gegebenheiten die negativ bewertet werden :

- jeder nicht besetzter Soll- Dienst führt zu einem Minuspunkt
- Planung entgegen ausdrücklichen Wunsch des Mitarbeiters
- Planung entgegen der Vorliebe „ungern Dienst“
- Planung entgegen Wunsch „keinen Dienst“

Gegebenheiten die positiv bewertet werden :

- für jeden nach Mitarbeiterwunsch entsprochener Dienstgabe
- sobald ein Schwellwert bei den Urlaubstagen im Monat überschritten wird, werden alle weiteren Urlaubstage mit je einem Punkt gutgeschrieben (Hintergrund : sind viele Mitarbeiter im Urlaub, ist es entsprechend schwieriger die Wünsche der Mitarbeiter umzusetzten)

CAC ist somit ein komplett mitarbeiterorientiertes Verfahren zur Bewertung eines Plans. Das Ergebnis wird in eine ASCII-Datei geschrieben, die in der Zelle C146 der DEFINITIONS- Tabelle mitsamt Dateipfad festgelegt wird. In der Zelle C145 wird der Pfad zum Texteditor (Standardeinstellung ist für W9x/ME/XP/ 2003 der Pfad C:\WINDOWS\NOTEPAD.EXE) angegeben, sollte der eingesetzte Rechner mit Windows NT 4.0 Workstation / Server oder mit Windows 2000 in allen Ausprägungen betrieben werden, so ist das Wort WINDOWS gegen WINNT auszutauschen. Damit das Schreiben in die Zelle funktioniert, sind Schreibrechte im Zielpfad erforderlich. Kontaktieren Sie gegebenenfalls ihre lokale IT-Administration oder korrigieren Sie den Pfad entsprechend dem Ordner „Eigene Dateien", in dem Sie höchstwahrscheinlich Schreibrechte besitzen.

Die Ausgabe des CAC erscheint in einem separaten Fenster:

```
Plananalyse vom      : (Datum) (Uhrzeit)
Erzielter Punktwert : -225
Fehlende Dienste in : A B C E W
Anzahl Urlaubstage..: 0
------------------------------------------------------------
Mitarbeiter.........: Müller, M.

Mitarbeiter.........: Schmid, A.
```

Unterhalb der Mitarbeiternamen würden „Wunschverstöße" angezeigt werden. Fehlende Dienste finden sich im Kopfbereich, ebenso die Anzahl aller gefundenen Urlaubstage in diesem Plan. Ebenfalls mit in die Berechnung fließt ein, ob die Werte des Stundensolls mit den Werten der Ist-Stunden innerhalb der in der DEFINITIONEN-Tabelle festgelegten Parameter liegen.

Erscheint beim Aufruf der qualitativen Analyse über die Schaltfläche „CAC" der Fehlerdialog „Laufzeitfehler 76" so ist der Parameter für die ASCII-Ausgabedatei nicht auffindbar – beispielsweise existiert der gewünschte Ordner, oder Pfad überhaupt nicht.

14 Parametrierung

In der DEFINITIONEN-Tabelle werden die individuellen Einstellungen für Ihre Einrichtung und Station hinterlegt. Die DEFINTIONS-Tabelle ist ein sehr sensibler Be

reich, der bei Fehleinträgen zu seltsamen oder falschen Ergebnissen führen kann. Die Anzahl von Parameter ist für die Flexibilität des Dienstplan-Modells notwendig.

Vor den einleitenden Worten zur Bedienungsanleitung findet sich eine Check-Liste für die Ersteinrichtung. Sollten Sie den Plan außerhalb des Ursprungshauses in Kaufbeuren einsetzen, wird die Verwendung der Check-Liste dringend empfohlen, da einige Parameter voneinander abhängig sind.

Version :	3.00 (B290120041) RC1

Die Versionsanzeige sollte nicht vom Benutzer verändert werden, da automatisierte Übernahmeprozesse (sprich Kopiertools von einer älteren Version zu einer neuen) diesen Wert auslesen. Die Versionsnummer ist wie folgt zu interpretieren:

3.00 (Hauptversion, Nebenversion)
B290120041 (Build-Datum : 29.01.2004, erste Fassung des Tages)
RC1 (Release-Candidate, keine fehlerfreie Endversion)

Die Nebenversion wird sich im Laufe der Hauptversion steigern, je nachdem wie viele Fehler im Echtbetrieb noch gefunden werden. Über Inter- beziehungsweise Intranet kann auf eine neuere, fehlerbereinigte Version heruntergeladen werden. Änderungen im Funktionsumfang sind innerhalb der Version 3 aktuell nicht geplant. Die Sammlung von Vorschlägen für die Nachfolgerversion hat bereits begonnen.

Stationsname :	Test

Der Stationsname ist vom Benutzer einzugeben. Der hier hinterlegte Name wird auf dem Dienstplan angedruckt.

Frühdienst ‚A'	08:00	06:30 - 15:00
Frühdienst ‚B'	08:00	08:00 - 17:30
Spätdienst	08:00	12:30 - 21:00
Zwischendienst	06:00	06:30 - 12:30
Zusatzdienst	00:00	00:00 - 00:00
Nachtdienst	09:15	20:45 - 06:45
Urlaub	07:42	

Stundenwerte – unterteilt in Regulär und Wochenturnus. Im DPE3 werden insgesamt sechs Dienstformen unterschieden. Zwei Frühdienste (A/B), ein Spätdienst, ein Zwischendienst (D), ein Zusatzdienst (E) und ein Nachtdienst. Für jede Dienstform werden die Dienstdauer und die Dienstzeit hinterlegt. Die Unterscheidung zwischen Regulär und Wochenturnus ermöglicht die Darstellung beispielsweise von Mitarbeitern, die als Beamter angestellt sind. Der Wochenturnus bietet aktuell keine Unterscheidung bei der Soll-Dienstzeit, außer dem manuellen Eintrag.

Dienst-Bezeichnung :	Frühdienst ‚A'	A
	Frühdienst ‚B'	B
	Spätdienst	C
	Zwischendienst	D
	Zusatzdienst	E
	Nachtdienst	W
	Urlaub	U

Festlegung der Dienstbezeichnungen. Hinter dem Buchstaben des Dienstes, findet sich eine Aufzählung von 1 bis 7 die unter keinen Umständen verändert werden sollte. Wird die Dienstbezeichnung auf mehr als ein Zeichen verlängert, so ist ein weiterer Parameter in der DEFINITIONEN-Tabelle einzustellen.

Sonderparameter :	Bezeichungsstellen ermitteln	1

Grundsätzlich werden zwei Methoden unterschieden, wie die Sollstellung für den Plan geschehen soll. Im älteren Verfahren wurden nur Wochenende, Feiertag und Wochentag unterschieden. Mit der Version DPE3 wurde die Möglichkeit eingeführt, jedem Wochentag eine individuelle Sollstellung einzutragen.

Tagesplanung aktiviert !!!	--	2
	--	2
	--	2
	--	1
	--	0
Tagesplanung aktiviert !!!	--	2
	--	2
	--	0
	--	2
	--	0

Plansoll	(Tagesplanung = 0)	0

Über den Parameter C75 wird gesteuert, ob die Wochenplanung (altes Verfahren) oder Tagesplanung (neues Verfahren) angewendet werden soll. Je nach gewählten Verfahren, fehlen (wie in der obigen Abbildung) die Bezeichnungen. Nur die Werte aus dem gewählten Verfahren werden für die Sollstellung verwendet.

Besetzung (Montag) :	Früh (A) :	2
	Spät (C) :	2
	Wache :	2
	Zwischend ‚B' :	1
	Zusatz ‚E' :	0

Mit den Zahlenwerten wird die Mindestbesetzung (Sollwert) für den gewählten Tag und die jeweilige Schicht festgelegt. Nur wenn der Wochentag auf einen Feiertag fällt, so wird der Feiertag automatisch vorgezogen.

Mitarbeiter (Soll/Ist) :	AP	1
	AS	1
	FP	1
	FPK	1
	FS	1
	HEP	1
	KP	1
	KPH	1
	KPS	0
	KS	1
	SD	1
	SL	1
	SLV	1
	SON	0
	ZDL	0

Je nach Dienstverhältnis ist im Bereich C76-C90 festzulegen, ob ein Dienst der jeweiligen Dienstform als Dienst für die Sollstellung mitgezählt wird oder nicht. In der Standardauslieferung zählen Krankenpflegeschüler, Krankenpflegeschülerinnen (beide als KPS definiert), sonstige Mitarbeiter (SON) oder Zivildienstleistende (ZDL) nicht mit bei der Erfüllung der Sollstellung. Die Zahl 0 repräsentiert das nicht mitzählen, die Zahl 1 das Gegenteil. Die Dienstformbenennung ist in DPE3 fest verankert und lässt sich hier nicht verändern.

Soll/Ist Abweichung Plus (in %)	15
Soll/Ist Abweichung Minus (in %)	15

Die Abweichung vom Sollparameter in Prozent nach Oben oder Unten steuert die farbliche Veränderung des erzielten Stundenwerts des einzelnen Mitarbeiters. Bei Unterschreitung bleibt die Zahl rot – bei Überschreitung wird die Zahl lila und doppelt unterstrichen.

Wochendienst VZ Multiplikator	1

Der Wochendienst VZ Multiplikator steuert die Sollstunden im Verhältnis zur Vollzeitkraft aus dem regulären Dienst. Ist der Mitarbeiter im Wochenturnus beispielsweise in einer Teilzeitbeschäftigung von 75% der Regelarbeitszeit eingestellt, so ist hier der Multiplikator auf 0,75 zu stellen.

Zeichen im Plan	!!!	Standard : !!!
SD1 - Stundenwert (kein Hochkomma) + Titel	01:45	Zeitungsgruppe (10:00 - 11:45)
SD2 - Stundenwert (kein Hochkomma) + Titel	09:15	Unterricht (Ganzer Tag)
SD3 - Stundenwert (kein Hochkomma) + Titel	02:15	Kochgruppe (16:00 - 18:15)

Es lassen sich insgesamt neun Sonderdienstformen fest parametrieren. In der obersten Zeile ist zunächst das Dienstplansymbol für die Sonderdienste und frei definierbaren Zusatzdienste festzulegen. Den Sonderdiensten sind neben der Dienstdauer der Name und der Zeitraum einzugeben. Die Formatierung der Eingabe ist der obigen Abbildung zu entnehmen.

Praktischerweise sollten die Sonderdienste von 9 rückwärts befüllt werden, da sie sich so besser im Planungshelfer aus dem Kombinationsfeld auswählen lassen.

Planungshelfer :	Malwert I (sollte stets auf 1 stehen)	1
	Malwert II	2
	Malwert III	3
	Malwert IV	4
	Malwert V	5
Mitarbeitereigenschaften :	gar keinen Dienst.	0
	ungern Dienst.	1
	gern Dienst.	2
	am liebsten Dienst.	3
	ungern Dienst :	4
	gern Dienst :	5
	Wochentagnamen	Sonntags
	Wochentagnamen	Montags
	Wochentagnamen	Dienstags
	Wochentagnamen	Mittwochs
	Wochentagnamen	Donnerstags
	Wochentagnamen	Freitags
	Wochentagnamen	Samstags
	Wochentagnamen	Feiertags

Alle Parameter aus der obigen Abbildung bedürfen in der Regel keiner Veränderung. Die Malwerte des Planungshelfers geben die Wiederholung der gewünschten Standarddienstform an. Unter den Mitarbeitereigenschaften sind die Wünsche entsprechend einer Zahl zugeordnet. Dieser Block steuert in erster

Linie die grafische Oberfläche. Innerhalb des VBA-Programmcodes wird fest auf die Zahlenwerte zugegriffen, sodass eine Änderung innerhalb der Parametrierung zu Programmfehlern führen wird.

Sonderparameter II :	Bezeichnung für Unterschrift 1	Unterschrift Stationsleitung / Datum
	Bezeichnung für Unterschrift 2	Unterschrift Bereichsleitung / Datum
	Nachtwachen anzeigen	1
	Minimalwert Zeiteintrag	0:00
	Maximalwert Zeiteintrag	200:00

Die Sonderparameter II steuern die Beschriftung unterhalb der beiden Unterschriftenfelder. Die Eingabe von Plusstunden in der Spalte AI wird durch eine Gültigkeitsprüfung überwacht, deren Minimal- und Maximalwert hier. Ob der Nachtwachenmerker verwendet werden soll wird hier festgelegt.

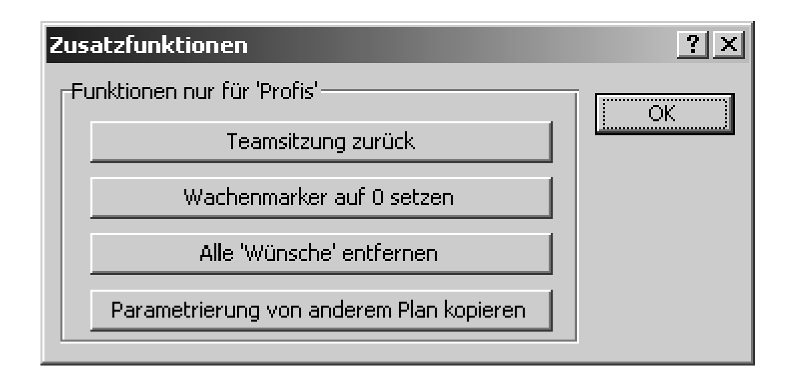
14.1 Parameter von anderem Plan kopieren

Die Parametrierung des Dienstplan in Excel ist ein komplexer Vorgang, bei dem sich viele Fehler einschleichen können. Unter Parametrierung versteht man die Einrichtung der Mitarbeiter, Dienstzeiten, Solldienste, Dienstlängen etc..

Als besonders lästig erscheint dem Anwender die wiederholte Parametrierung aufgrund eines Versionswechsels. Mit einer neuen Funktion im Menüpunkt „Diverses“ soll die Parametrierung aus der Version III in die Version 3.10 übernommen werden können.

Ausnahme ist der Planinhalt und die Einstellungen zum Punktesystem des CAC – diese Werte müssen von Hand nachgeprüft werden. Führen Sie folgende Schritte durch, um die Parameter von der Version III nach 3.10 zu übertragen:

- Speicherung der alten Version auf Laufwerk C mit der Bezeichnung ALT.XLS
- Speicherung der neuen, leeren 3.x0 Version auf dem Desktop als NEU.XLS
- in NEU.XLS ist das Menü DIV anzuklicken
- mit dem Befehl „Parametrierung von anderem Plan kopieren“ wird der Vorgang gestartet

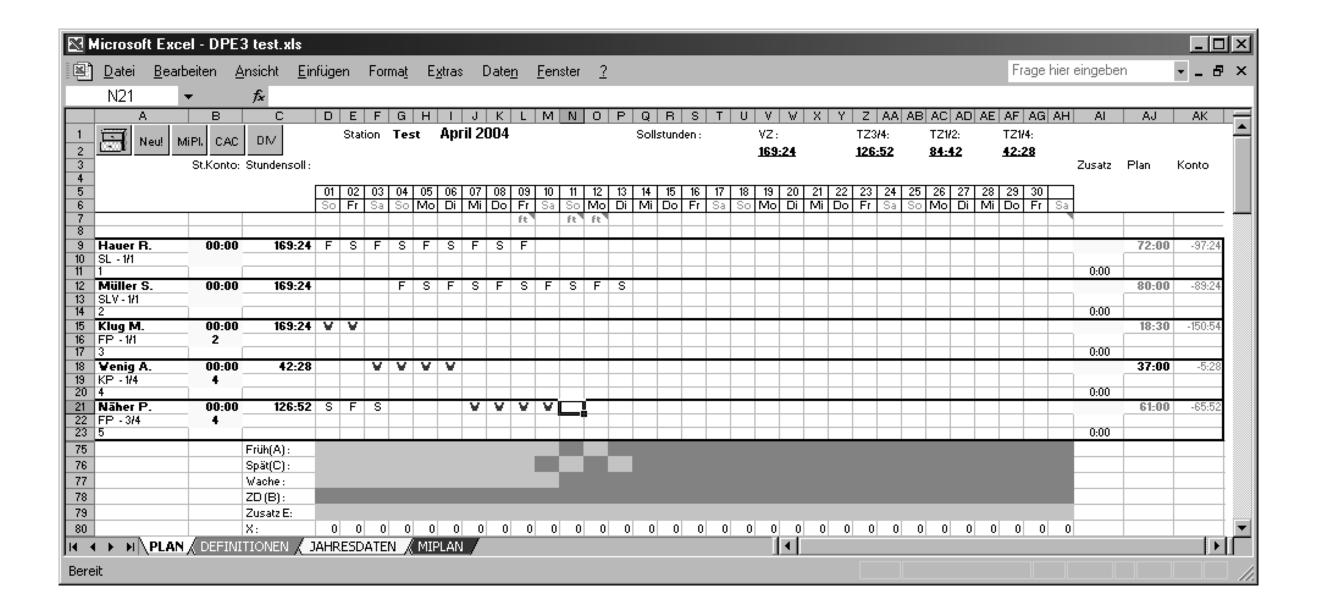


Beachten Sie, dass eine Übernahme der Parameter aus der Version 2.x nur auf die Namen und Anstellungen der Mitarbeiter beschränkt ist. Weitere Informationen finden sich im Quelltext von Modul 3 para_uebernahme().

15 Nutzungsempfehlung

In der gesamten bisherigen Dokumentation wurde viel über die Technik in Excel, der Programmierung in VBA und der Grundeinrichtung des fertigen DPE3 gesprochen. Doch wie sich nun ein Dienstplan mit dem Tool erstellen lässt, wurde mit noch keinem Wort erwähnt – dies wird mit diesem Kapitel nachgeholt.

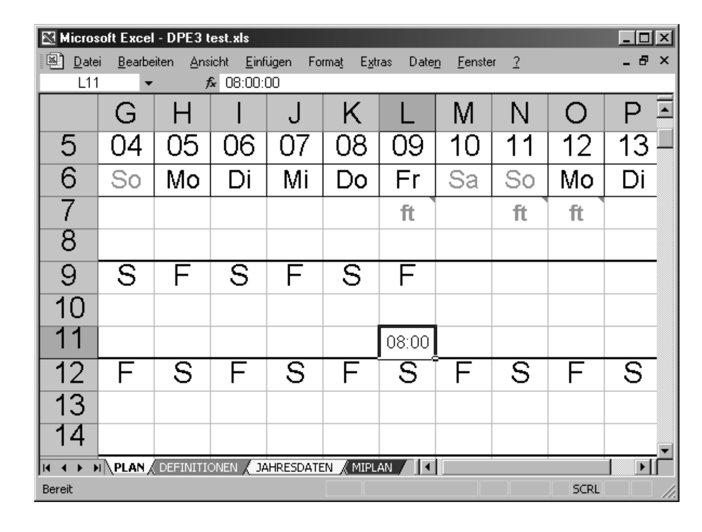
Die Bedienung des Dienstplan in Excel richtet sich nach der Parametrierung, sodass eine generelle Nutzungsanleitung kaum erstellt werden kann. Im Folgenden wird davon ausgegangen das die virtuelle Einrichtung über einen Frühdienst (F), einen Zwischendienst (Z), einen Spätdienst (S) und einen Nachtdienst (N) verfügt. Jede Dienstform ist 8 ½ Stunden lang, sprich acht Arbeitsstunden. Für jede Dienstform wird eine Sollbesetzung von einer Kraft angenommen. Die folgenden Bildschirmdarstellungen wurden bewusst gekürzt.



Für jeden eingetragenen Dienst wird der hinterlegte Stundenwert von 8:00 angerechnet und der Dienst in der Sollstellung am unteren Tabellenbereich grün eingefärbt. Rote Kästchen stehen für unterbesetzten Dienst, grüne Kästchen für ausreichend besetzte Schichten und orange-farbene Kästchen für über Bedarf geplante Dienste. Ein „ordentlich“ besetzter Dienstplan hat somit in der Soll/Ist-Prüfung keine roten Bereiche.

15.1 Eingabe : Feiertage

Dem aufmerksamen Leser wird in der obigen Abbildung nicht entgangen sein, dass der 09.04.04 ein Feiertag ist und – je nach Regelung in der Einrichtung – mit einer Feiertagsvergütung zu rechnen ist. Je nach Auslegung des BAT ist für den Freitag ein FF (Feiertagsfrei mit Stundenanrechnung) an einem anderen Tag des Plans zu gewähren. In anderen Einrichtungen wird die „Stundenverdoppelung“ am gleichen Tag eingetragen. Da es hier unterschiedliche Ausprägungen gibt, bietet DPE3 derzeit keine automatisierte Berechnung des Feiertags. Es ist der Dienstplaner selbst gefragt.



Wird in der dritten Zeile des betreffenden Tages ein Stundenwert eingegeben, hier wurde 8:00 eingetragen, so wird der Tag entsprechend mit 16:00 Stunden kalkuliert.

Wie auch immer die Eingabe durchgeführt wird : Der Stundenwert muss in der dritten Zeile manuell durch den Benutzer eingegeben werden. Die Eingabe erfolgt, wie im DPE3 üblich, stets im Format hhh:mm -> maximal dreistellig der Stundenwert gefolgt von einem Doppelpunkt mit anschließender zweistelliger Angabe der Minuten. Wird ein unzulässiger Zeitwert eingegeben, so erscheint eine dezimale Anzeige des Stun
denwerts. Die schlichte Eingabe einer Eins würde von der Tabelle als 24 Stunden (ein Tag) interpretiert werden. Eine Prüfung der Eingabe findet aktuell nicht statt.

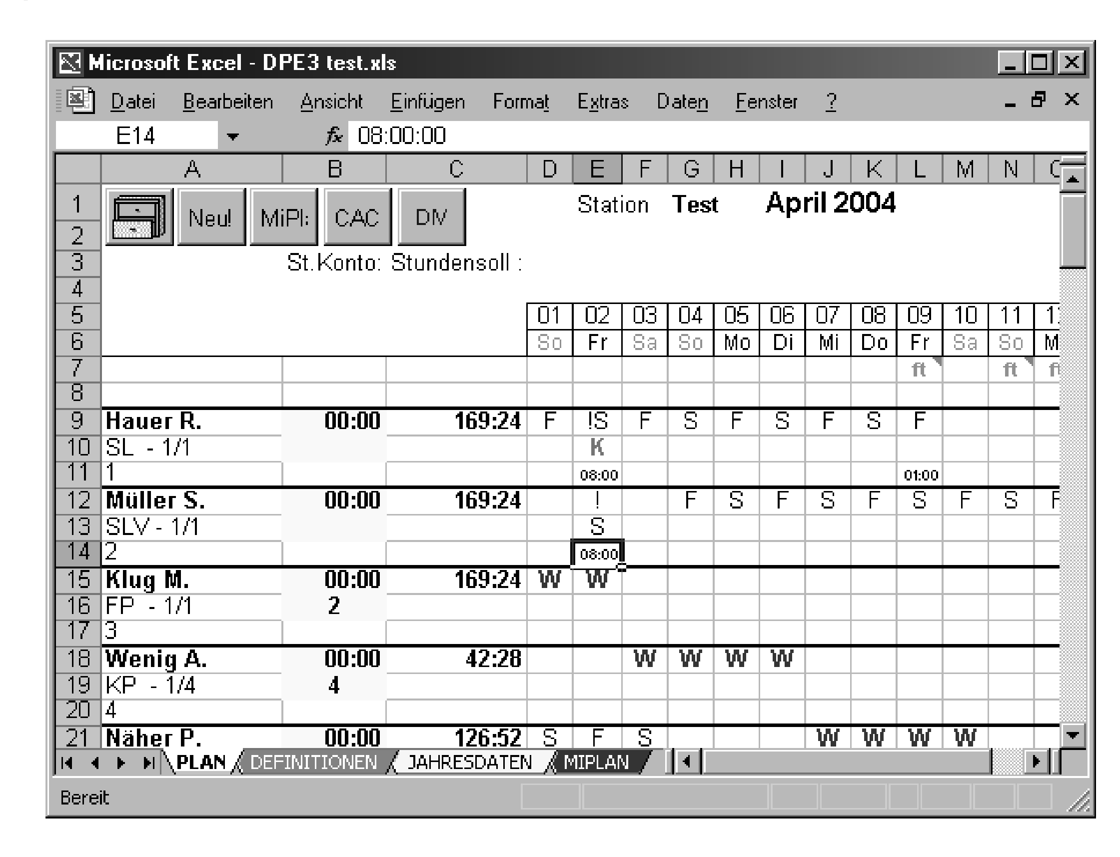
15.2 Planung, Verlauf und Krankheit

Der Buchstabe „K“ ist im DPE3 für die Eingabe eines Krankheitstags reserviert und wird stets entsprechend dem Urlaubstag berechnet – also auch entsprechend dem Einstellungsverhältnis (Voll- oder Teilzeit). Zu unterscheiden ist hier das nicht geplante Krank. Bei unvorhersehbarer Krankheit, bei der ein Mitarbeiter seinen geplanten Dienst nicht antreten kann, wird der Stundenwert der eigentlich geplanten Dienstform verwendet. Dies ist aber nur dann von Bedeutung, sofern der DPE3 auch für die Erfassung der tatsächlich geleisteten Arbeit genutzt wird. Bei der einfachen Planung – sprich der Vorbereitung des Plans für einen zukünftigen Monat – kann ein Krank in der Laufzeit nicht vorkommen.

Wird DPE3 aber für die Erfassung des tatsächlichen Monatsverlaufs genutzt, so sind eigentlich zwei Dateien für einen Monat vorhanden. Die erste Datei mit der ursprünglichen Planung, die ausgedruckt den Mitarbeitern ausgehändigt wird und die spätere Korrektur mit dem tatsächlichen Verlauf.

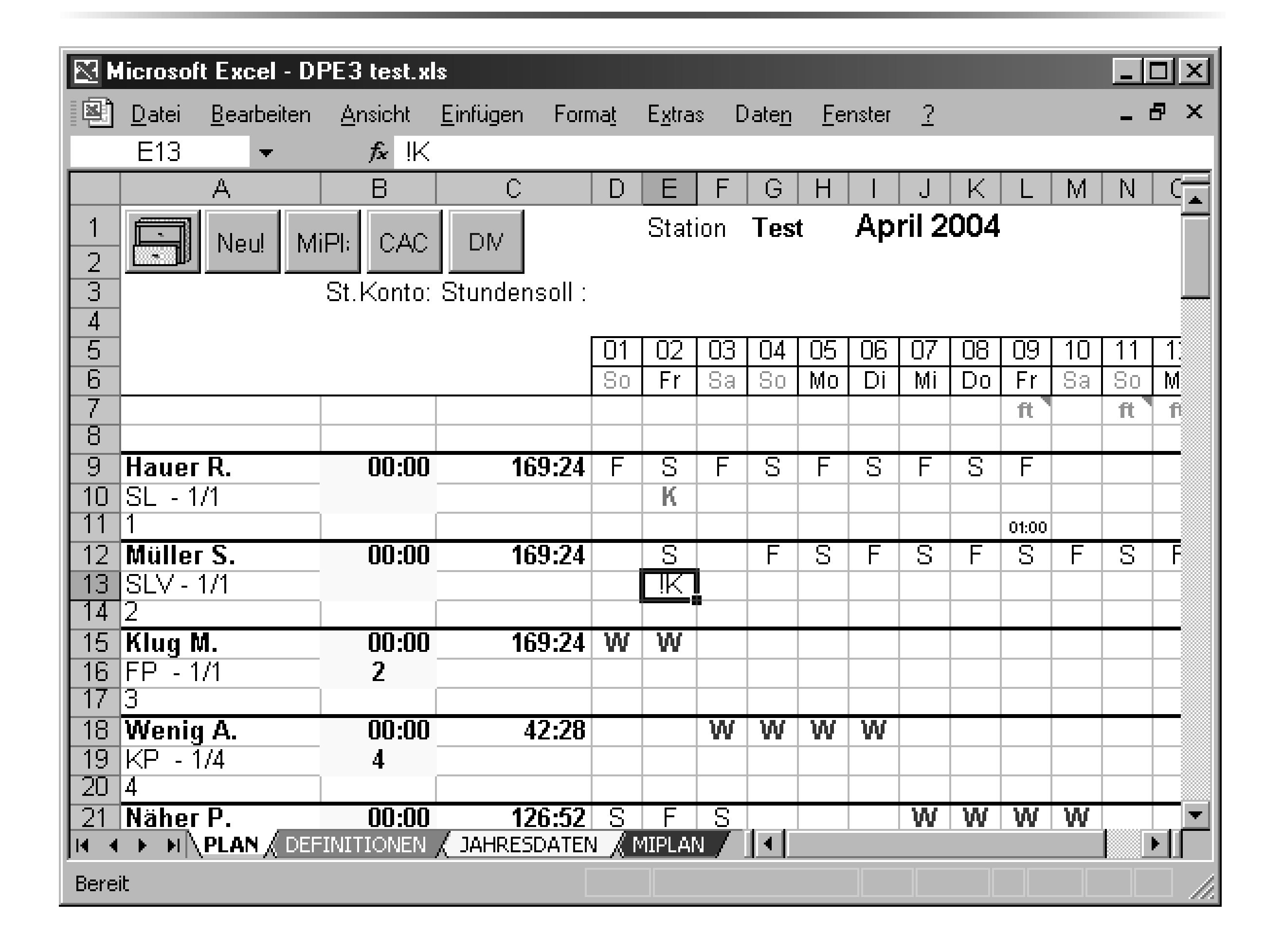
Wie aber sollen die Veränderungen am Plan festgehalten werden? Diese Frage ist wieder sehr individuell dem Benutzer überlassen. Am Bezirkskrankenhaus Kaufbeuren, der Einrichtung, die als erstes DPE3 einsetzte, entschied man sich für eine Entwertung des nicht genommenen Diensts mit Hilfe eines Ausrufezeichens. Durch das vorgelagerte Ausrufezeichen bleibt erkennbar, welche Dienstform ursprünglich vom Dienstplaner vorgesehen war.

Beispiel :



Mitarbeiter Hauer war am Freitag den 02.04.2004 im Krankenstand und konnte aus diesem Grund den Spätdienst nicht wahrnehmen. Durch das vorgelagerte Ausrufezeichen vor dem „S“ des Spätdiensts wird die Stundenberechnung für diesen Dienst außer Kraft gesetzt. In der Zeile zwei, die jeden beliebigen Wert annehmen kann, wird das „K“ für Krankheit eingetragen. Für Krankheit, Urlaub und Nachtdienst „W“ ist eine bedingte Formatierung hinterlegt, die für die Formatänderung verantwortlich ist. Die 08:00 in der dritten Zeile sorgen für eine Anrechnung der Krankheitszeit. Mitarbeiter Müller übernahm den Dienst, darum ein freies Ausrufezeichen in der ersten Zeile – geplant war kein Dienst. In der Zeile zwei wird die durchgeführte Dienstform und in der Zeile drei die dazugehörigen Stunden eingetragen.

Nachteil dieses Vorgehens ist sicherlich die notwendige Eingabe vieler Stundenwerte. Gleichzeitig ermöglicht dies die Anpassung an die Gegebenheiten der Station oder Abteilung. Eine einfachere Art wäre den ursprünglich geplanten Dienst nicht zu entwerten und dem „einspringenden“ Mitarbeiter in der Zeile zwei zu markieren:



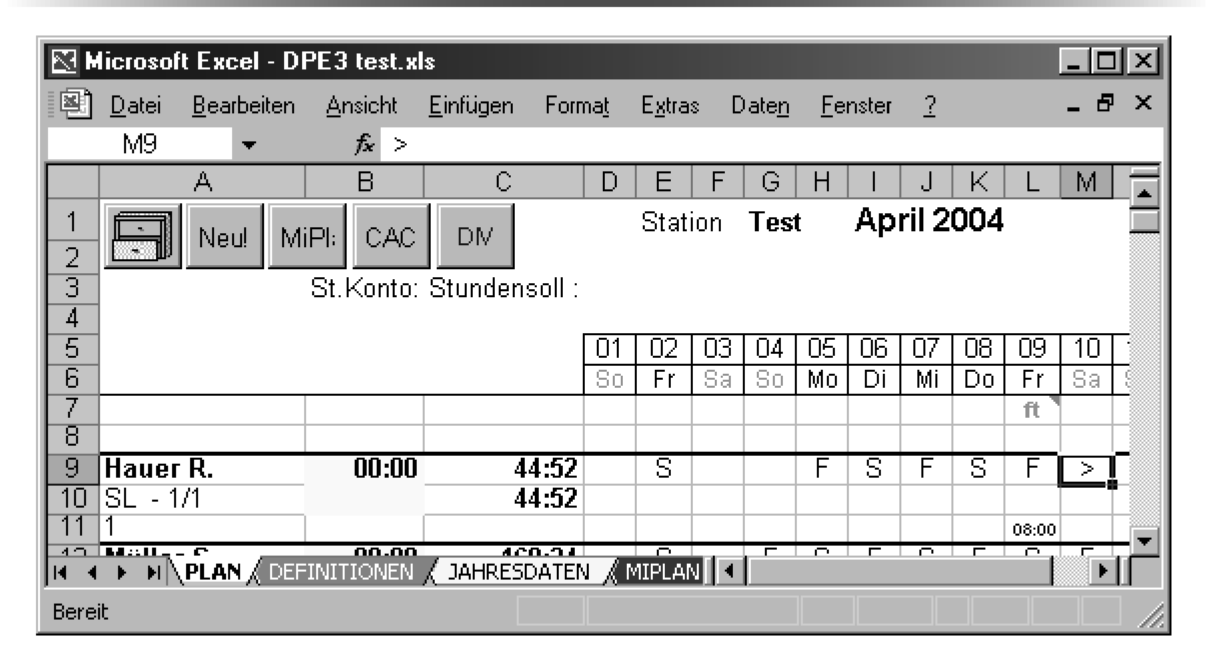
Dies erspart die Eingabe der Stundenwerte und führt zum gleichen Ergebnis, sofern für die Krankheitswerte nicht unterschiedliche Stundenwerte benutzt werden, als für die eigentlich geplante Dienstform. Die zuvor benannte Methode bildet zudem die Zusatzstunden in der Spalte AI ab, was eventuell auch Beachtung finden könnte.

Welche Methode Anwendung findet ist jedem Benutzer selbst überlassen – empfehlenswert ist sicherlich die einheitliche Nutzung innerhalb des Unternehmensstandorts.

15.3 Abweichendes Stundensoll

Nicht immer haben Mitarbeiter exakt das monatliche Stundensoll, was durch DPE3 vorgegeben werden kann. Die Begründungen können sehr unterschiedlich sein. Einerseits gibt es vermehrt Verträge in den Einrichtungen, die sich sehr individuell an die Wünsche des Mitarbeiters oder des Unternehmens anlehnen – beispielsweise ein 12 Stunden pro Woche Vertrag. Anderseits kann es auch sein, dass ein Mitarbeiter nur den halben Monat auf der Station planbar ist – beispielsweise Schüler.

Das Feld unterhalb des generierten Monatsstundensoll wird für eine manuelle Eingabe des Sollstundenwerts für den Planungsmonat verwendet. Beispielsweise verlässt der Mitarbeiter in Kürze die Station:

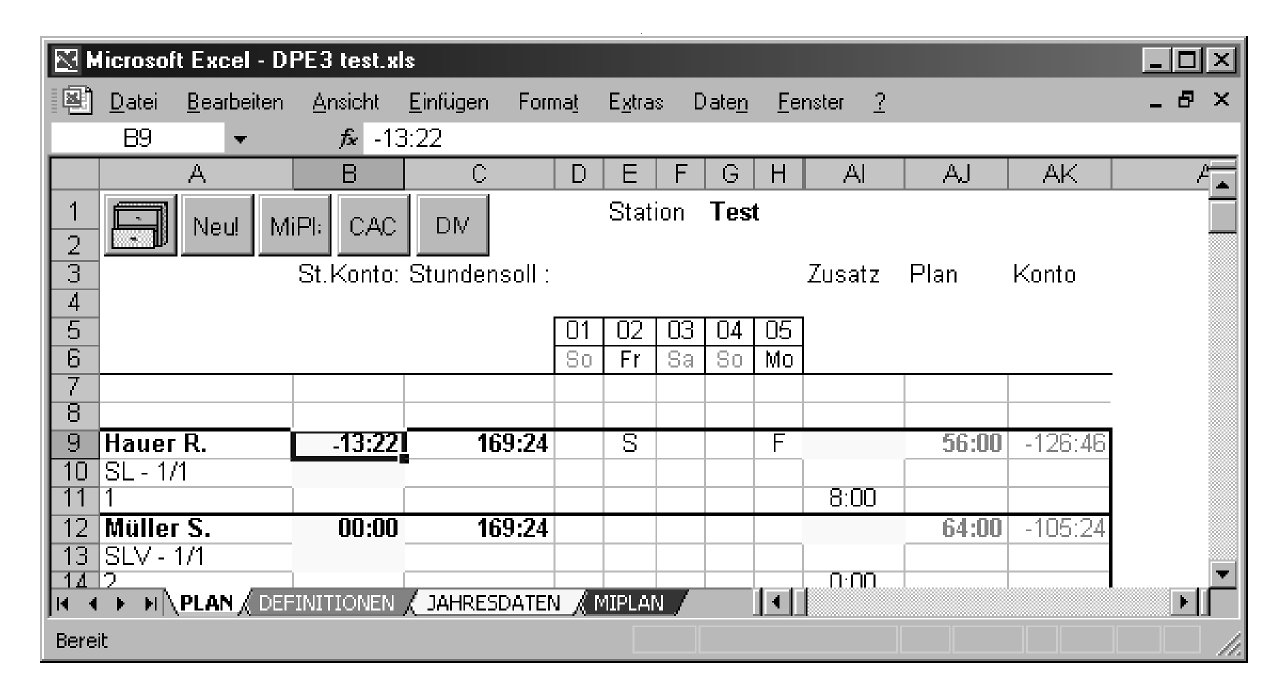


Eigentlich wird Mitarbeiter Hauer im Unternehmen im Vollzeitvertrag geführt, erkennbar an dem „1/1" hinter der Dienstbezeichnung. Da aber am 10.04.2004 die Dienstzugehörigkeit auf der Planungsstation endet, hier symbolisiert durch das Zeichen „Größer als", sind auf dieser Station nur 44:52 Stunden abzuleisten. Da in der Zelle C10 dieser Wert eingetragen ist, wird auch in C9 der Standard für Vollzeitkräfte übergangen und der manuelle Wert des Dienstplaners verwendet.

15.4 Plus- und Minusstunden

Je nachdem ob die Einrichtung ein Stundenkonto führt, so bietet DPE3 die Möglichkeit an, den Stundenkontowert im Dienstplan einzutragen. Grundsätzlich wird der Wert für die Kalkulation des sich nach diesem Planungsmonat anschließenden Stundenkontowerts verwendet, nicht aber für die Berechnung des laufenden Monats. Sprich der laufende Planungsmonat geht vor – es wird die möglichst nahe Planung an das Stundensoll des Monats angepeilt (siehe qualitative Plananalyse).

Die Eingabe in Spalte B ist sowohl in Plus- als auch Minusstunden möglich. Handelt es sich um Überstunden, die der Mitarbeiter auf seinem Stundenkonto führt, so sind diese durch die einfache Eingabe des Zahlenwert im Format hh:mm zu hinterlegen. Minusstunden sind mit einem führenden Minuszeichen im Format –hh:mm einzugeben.



Durch die Eingabe des Stundenkontowert -13:22 wird der künftigen Stundenkontowert (Spalte AK) um eben diesen Wert gesenkt. Auf den Wert des aktuellen Plans hat dieser Eintrag jedoch keinerlei Auswirkung.

Die farblichen Veränderungen in den Spalten AJ und AK sind abhängig vom Parameter „Soll/Ist Abweichung Plus (in %)“ und „Soll/Ist Abweichung Minus (in %)“ (siehe Parametrierungsbeschreibung).

15.5 Eingabe von Sonderdiensten

Die Methode der Eingabe von Sonderdiensten ist noch relativ jung und wurde mit der Version DPE3 eingeführt. Durch diese Dienste soll die Möglichkeit gegeben werden sehr individuelle Anforderungen problemlos umzusetzen. Anstelle eines Buchstabensymbols in der Dienstplanübersicht erscheint in der Standardeinstellung der Eintrag „!!!“. Wird der Mauszeiger auf dem Dreifachausrufezeichen positioniert erscheint in einer kleinen „Sprechblase“ der Inhalt des geplanten Dienstes.

Sowohl die Sonderdienste als auch der Eintrag im Dienstplan selbst lassen sich in der Tabelle DEFINITIONEN den eigenen Wünschen entsprechend anpassen. Um eine Sonderdienstform oder eine freie definierte Dienstform wieder komplett aus dem Plan zu löschen ist im Planungshelfer „Kein Dienst“ anzuklicken. Werden einfach nur die Einträge manuell gelöscht, verbleiben Eintragsreste im so genannten „Schattenplan“, der als Basis für den Mitarbeiterplan mitgenutzt wird.



Sonderdienste und frei definierte Dienstformen können nicht direkt über die Tastatur eingegeben werden, sondern über den Planungshelfer. Dieses Dialogfeld wird mit einem Doppelklick auf dem gewünschten Tag in der ersten Zeile des jeweiligen Mitarbeiters aktiviert.

Soll zusätzlich zum Sonderdienst oder frei definierten Dienst ein Standarddienst eingeplant werden, so ist dies möglich, indem zunächst der Sonderdienst oder frei definierte Dienst eingetragen wird. Das Fenster des Planungshelfers ist mit „OK“ zu verlassen.
Der Stundenwert für den Sonder- beziehungsweise frei definierten Dienst wird in die dritte Zeile des jeweiligen Mitarbeiters eingetragen. Die Bezeichnung für den Dienst wird in einem „Schattenplan“ abgelegt, der auch nach dem Eintrag des Standarddiensts erhalten bleibt.

Nach der Eingabe des Sonder- beziehungsweise frei definierten Dienst wird einfach an dem gewünschten Tag der Buchstabe des Standarddienst über die !!! eingetragen.

Planungshelfer
Standarddienstform
[keinen Dienst] x 1
OK
Keinen Dienst
Sonderdienstform
[keinen Dienst]
Freie Definition
Titel mit Zeitangabe
Zeitwert (hh:mm) 00:00
Titel (HH:MM - HH:MM)
Aus den Wünschen

Bei der Eingabe von freien Dienstformen ist das Eingabeformat zu beachten:

Patientenbegleitung (13:00 - 15:00)

Findet DPE3 im hinteren Bereich keinen abgeschlossenen Bereich mit zwei runden Klammern, so wird der Eintrag wohlmöglich auf dem Mitarbeiterplan nicht korrekt angedruckt. Der Zeitwert ist der Wert, der in die dritte Zeile des jeweils zu planenden Tags eingetragen wird.

16 Ausblick : Künftige Funktionalitäten

Es wird sicherlich Leser geben, die sich fragen, warum denn in dem hier gezeigten Modell diese und jene Funktionalität von Microsoft Excel nicht angewendet wird oder warum sich so einfach umzusetzende Hilfsmethoden in der Planung – beispielsweise die Anzeige nicht gewünschter aber benötigter Dienstformen über die Statusleiste von Microsoft Excel – nicht finden. Der Grund ist ganz einfach – es wurde bis dato nicht umgesetzt und irgendwann war einfach ein Termin, an dem hieß es Version III des Planers soll eingeführt werden. Bereits während dieser Text geschrieben wird, ist die Planung für die Version 3.2 auf dem Tisch. Das wird für den Leser dieses Buchs auch kein Nachteil sein, denn die vorgestellten Konzepte und die Methode der Dienstplanung mit Microsoft Excel werden durch die Erweiterung von Funktionen nicht außer Kraft gesetzt.

Die jeweils gültige Fassung des Dienstplan in Excel findet sich im Internet auf der Homepage des Pflegeinstituts unter www.pflegeinstitut.de/dpl. Da es sich um ein Projekt gemäß der GNU General Public License handelt, können Sie als Benutzer mit dem Konzeptentwurf machen was Sie wollen, solange ein rechtlicher Hinweis auf den Autoren erhalten bleibt.

Für den Einsatz in der Psychiatrie wurden die Fähigkeit zur Eingabe individueller Dienstformen und der Ausdruck eines Mitarbeiterplans als ersten Schritt in eine andere Form der Dienstgestaltung gewählt. Die Krankenpflege wird sich im Verlauf der nächsten Jahre stark verändern – auch außerhalb der psychiatrischen Einrichtungen. In den Mittelpunkt der Arbeit wird vermehrt der Patient gerückt – ein zweiter Mittelpunkt wird sich aus den individuellen Fähigkeiten der einzelnen Mitarbeiter bilden, die über die Planung innerhalb einer einzelnen Station wohlmöglich hinausgehen. Wie lange es sich Einrichtungen leisten werden, auf jeder Station Mitarbeiter mit ähnlichen Fähigkeiten und Funktionen vorzuhalten ist fraglich. Der Autor ist der Meinung, dass eine künftige Terminplanung von Gruppen, die durch Pflegende durchgeführt werden und Gesprächstermine mit dem Patienten, zumindest in den psychiatrischen Einrichtungen, in absehbarer Zukunft außerhalb der eigentlichen Dienstplanung geschehen wird. Mit dem DPE3 wird der erste bescheidene Versuch unternommen, diese neue Anforderung umzusetzen.

Es wäre kein Problem eine weitreichende Terminplanung mit der herkömmlichen Dienstplanung zu verbinden, nicht jedoch durch den Einsatz einer einzelstehenden Microsoft Excel Datei. Auch würden sich Parametrierungsdaten viel einfach in der Windows-Registrierdatenbank ablegen lassen oder in einer Art klassischer .INI-Datei. Der Grund, warum diese Ideen noch nicht umgesetzt wurden ? Didaktisch ist es zunächst besser nur eine einzelne Datei zu haben, in der verschiedene Programmtechniken unter Excel vorgestellt werden. Ein weiterer Grund ist das raschere Einführungstempo als wenn auf teurere Programme wie Microsoft Access gesetzt wird, die zunächst bewilligt, beschafft und erlernt werden müssen, als wenn eine Standardapplikation wie Microsoft Excel verwendet wird. Und – das ist vielleicht sogar der Hauptgrund – ist dann eine individuelle Veränderung durch Sie, als Leser, nicht mehr so einfach möglich, da diese Programme wesentlich schwieriger zu erlernen sind.

Unabhängig von diesen Gedanken ist sicherlich eine Version für Open Office wünschenswert, um auch die Plattformen nutzen zu können, die nicht von Microsoft mit einem Office-Paket ausgestattet werden. Eine Schnittstelle zu Microsoft Outlook, welches in vielen Stationen im Gesundheitswesen verwendet wird, wäre ebenfalls eine nette Funktion. Die Loslösung von den fest eingetragenen Stundenwerten mit der Möglichkeit aufgrund individueller Verträge die Sollstunden abzubilden. Auch dies wäre eine sinnvolle Funktion. Bei der Visualisierung gibt es viele andere Möglichkeiten, die den Arbeitsfluss verbessern könnten. Prüfungsroutinen, die Eingaben auf ihre Sinnhaftigkeit hin prüfen – wären sicherlich ebenfalls sinnvoll. Im VBA-Quellcode könnte anstelle immer auf die direkte Adresse ein Namensbezug verwendet werden, das würde es ermöglichen, die Parametrierungstabelle aufzuräumen. Aus Performancegründen wäre es ebenfalls sinnig den Planinhalt zunächst in ein Array einzuladen und dann innerhalb dieses Array die Berechnungen anzustellen…

… merken Sie was? Das können Sie jederzeit tun – oder Sie schauen regelmäßig auf der Homepage des Pflegeinstituts vorbei, vielleicht gibt's ja schon eine neuere Fassung, die Sie mit Hilfe der Funktion „Parameter von anderem Plan kopieren“ einführen. Oder Sie nutzen das Onlineforum auf der gleichen Homepage um mit anderen Nutzern und dem Autor über Veränderungen für künftige Versionen zu diskutieren.